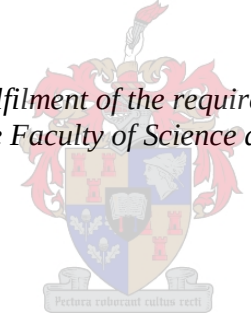
A qualitative model of evolutionary algorithms

by
Francois Fagan



Thesis presented in fulfilment of the requirements for the degree of Master of Science in the Faculty of Science at Stellenbosch University

Supervisor: Professor J H van Vuuren

April 2014

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

April 2014

Abstract

Evolutionary Algorithms (EAs) are stochastic techniques, based on the idea of biological evolution, for finding near-optimal solutions to optimisation problems. Due to their generality and computational speed, they have been applied very successfully in a wide range of disciplines. However, as a consequence of their stochasticity and generality, very little has been rigorously established about their performance. Developing models for explaining and predicting algorithmic performance is, in fact, one of the most important challenges facing the field of optimisation. A qualitative version of such a model of EAs is developed in this thesis.

There are two paradigms for explaining why EAs are expected to converge toward an optimum. The traditional explanation is that of Universal Darwinism, but an alternative explanation is that they are hill climbing algorithms which utilise all possible escape strategies — restarting local search, stochastic search and acceptance of non-improving solutions. The combination of the hill climbing property and the above escape strategies leads to a fast algorithm that is able to avoid premature convergence.

Due to the difficulty in mathematically or empirically explaining the performance of EAs, terms such as *exploitation*, *exploration*, *intensity* and *diversity* are routinely employed for this purpose. Six prevalent views on exploitation and exploration are identified in the literature, each expressing a different facet of these notions. The coherence of these views is substantiated by their deducibility from the proposed novel definitions of exploitation and exploration. This substantiation is based on a novel hypothetical construct, namely that of a *Probable Fitness Landscape* (PFL), which both unifies and clarifies the surrounding terminology and our understanding of the performance of EAs.

The PFL is developed into a qualitative model of EAs by extending it to the notion of an *Ideal Probability Distribution* (IPD). This notion, along with the criteria of diversity and computational speed, forms a method for judging the performance of EA operators. It is used to explain why the principal operators of EAs, namely mutation and selection, are effective.

There are three main types of EAs, namely *Genetic Algorithms* (GAs), Evolution Strategies and Evolutionary Programming, each of which employ their own unique operators. Important facets of the crossover operator (which is particular to GAs) are identified, such as: opposite step vectors, genetic drift and ellipsoidal parent-centred probability distributions with variance proportional to the distance between parents. The shape of the crossover probability distribution motivates a comparison with a novel continuous approximation of mutation, which reveals very similar underlying distributions, although for crossover the distribution is adaptive whereas for mutation it is fixed. The PFL and IPD are used to analyse the crossover operator, the results of which are contrasted with the traditional explanations of the Schema Theorem and Building Block Hypothesis as well as the Evolutionary Progress Principle and Genetic Repair Hypothesis. It emerges that the facetwise nature of the PFL extracts more sound conclusions than the other explanations which, falsely, attempt to prove GAs to be superior.

The use of facetwise and qualitative models are justified by their success in explaining the performance of EAs. It is argued that the best direction for EA research to progress is to refrain from competitive testing and attempts to model the so-called equations of motion, but to encourage the development of scientifically justifiable facetwise models of algorithmic performance.

Uittreksel

Evolusionêre Algoritmes (EAs) is stogastiese tegnieke vir die bepaling van naby-optimale oplossings vir optimeringsprobleme wat gebaseer is op die beginsel van biologiese evolusie. As gevolg van hul algemene toepasbaarheid en hoë berekeningspoed, is hierdie algoritmes al met groot sukses in 'n wye verskeidenheid dissiplines toegepas. Die stogastiese aard en algemene toepasbaarheid van hierdie klas van algoritmes het egter tot gevolg dat baie min al oor hul werkverrigting formeel bewys is. Die ontwikkeling van modelle waarmee die doeltreffendheid van algoritmes verklaar en voorspel kan word, is trouens een van die grootste uitdagings in die studieveld van optimering. 'n Kwalitatiewe weergawe van so 'n model word in hierdie verhandeling vir EAs daargestel.

Daar bestaan twee paradigmas vir die verklaring van waarom daar van EAs verwag word om na 'n optimum te konvergeer. Die tradisionele verklaring geskied aan die hand van Universele Darwinisme, maar 'n alternatiewe verklaring is dat hierdie algoritmes bergtop-soekend is en van alle moontlike ontsnapstrategieë gebruik maak — lokale soekstrategieë, stogastiese soekstrategieë en die aanvaarding van minderwaardige oplossings. Die kombinasie van die bergtop-soekende eienskap en die insluiting van die bogenoemde ontsnapstrategieë gee aanleiding tot vinnige algoritmes wat daartoe in staat is om voortydige konvergensie te vermy.

Omdat dit moeilik is om die werkverrigting van EAs wiskundig of empiries te verklaar, word terminologie soos *uitbuiting*, *verkenning*, *intensiteit* en *diversiteit* roetinegewys vir hierdie doel ingespan. Ses heersende menings in die literatuur oor uitbuiting en verkenning word geïdentifiseer wat elkeen 'n ander faset van hierdie begrippe uitlig. Die samehang van hierdie menings word deur hul afleibaarheid uit nuwe definisies van uitbuiting en verkenning gedemonstreer. Hierdie demonstrasie is gebaseer op 'n nuwe hipotetiese konstruk, naamlik dié van 'n *Waarskynlike Fiksheidslandskap* (WFL), wat beide die omliggende terminologieë en ons begrip van die werking van EAs enersyds verenig en andersyds verduidelik.

Die begrip van 'n WFL word tot 'n kwantitatiewe model vir EAs ontwikkel deur dit tot die konstruk van 'n *Ideale Waarskynlikheidsverdeling* (IWV) uit te brei. Hierdie konsep word saam met die kriteria van diversiteit en berekeningspoed gebruik om 'n metode te ontwikkel waarmee die werkverrigting van EAs beoordeel kan word. Die IWV word gebruik om te verklaar waarom die hoofoperatore van EAs, naamlik mutasie en seleksie, doeltreffend is.

Daar is drie tipes van EAs, naamlik *Genetiese Algoritmes* (GAs), *Evolusionêre Strategieë* en *Evolusionêre Programmering*, wat elk hul eie, unieke operatore bevat. Belangrike fasette van die oorgangsoperator (wat eie is aan GAs) word uitgelig, soos regoorstaande trapvektore, genetiese neiging en ellipsoïdale ouer-gesentreerde waarskynlikheidsverdelings met variansies wat eweredig is aan die afstand tussen ouers. Die vorm van die oorgangs-waarskynlikheidsverdeling gee aanleiding tot 'n vergelyking tussen die begrip van oorgang en 'n nuwe, kontinue benadering van mutasie. Daar word gevind dat die onderliggende verdelings baie soortgelyk is, alhoewel die oorgangsverdeling aanpasbaar is, terwyl die verdeling vir mutasie vas is. Die WFL en IWV

word gebruik om die oorgangsoperator te analiseer en die resultate van hierdie analise word teenoor die tradisionele verklarings van die Skemastelling en Boublok-hipotese sowel as die Evolusionêre Vooruitgangsbeginsel en die Genetiese Herstel-hipotese gekontrasteer. Dit blyk dat meer grondige gevolgtrekkings gemaak kan word uit die fasetgewyse aard van die WFL as uit ander verklarings wat valslik poog om die meer doeltreffende werkverrigting van GAs te demonstreer.

Die gebruik van faset-gewyse en kwalitatiewe modelle word geregverdig deur hul sukses in terme van die verklaring van EA werkverrigting. Die argument word gemaak dat die beste rigting vir voortgesette navorsing oor EAs is om weg te bly van vergelykende studies en die afleiding van sogenaamde vergelykings van beweging, maar om eerder die ontwikkeling van wetenskaplik-gefundeerde, faset-gewyse modelle vir algoritmiese werkverrigting na te streef.

Acknowledgements

The author wishes to acknowledge the following people for their various contributions towards the completion of this work:

- First and foremost, I am forever indebted to my supervisor, Prof Jan van Vuuren, for giving me the opportunity to pursue my academic interests. I deeply appreciate all of his guidance and wisdom that he has given me, both in my work and life in general. During the last eighteen months he has been the universe's chief conspiritor in helping me achieve my goals.
- I am grateful to the Department of Logistics for the use of their facilities, especially the postgraduate computer lab.
- I would like to thank all of the students in the lab whom I have had the pleasure of getting to know over the last two years. They were always warm and inviting, which helped me settle into the once scary Stellenbosch environment. I truly appreciate their friendship and support.
- The financial assistance of the University of Stellenbosch and the South African National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the NRF.
- Finally, I would like to thank those who are closest to me, my friends and family. In particular, my lakka girlfriend, Chiara Gaylard, and my ever supportive parents, Johan and Marion Fagan. This thesis is dedicated to my mother, so that she may have a university thesis of her own.

Table of Contents

Glossary

Individual A candidate solution to an optimisation problem.

Child A newly generated individual.

Parent An individual from which a child is generated.

Population A set of individuals, typically present during the same iteration.

Generation An iteration of an algorithm with a population.

List of Reserved Symbols

Symbol	Meaning
f	Fitness function
$\mathcal{S}$	Search space
$\|\| \cdot \|\|$	Standard Euclidean norm in $\mathbb{R}^n$
n	Number of dimensions
N	Number of iterations
$\hat{s}$	Global maximum
s'	Local maximum
s^*	Current best candidate solution
s^k	k^{th} indexed candidate solution
s_i	i^{th} coordinate of a candidate solution
Δs	Step vector
d	Direction vector
σ	Step size
T_m	Minimum temperature
(μ, λ)	EA with parent population size μ and offspring size $\lambda \geq \mu$, with selection only from the offspring population
$(\mu + \lambda)$	EA with parent population size μ and offspring size λ, with selection from the union of the parent and offspring populations
p_m	Mutation rate
p_c	Crossover probability
$G(\mu, \sigma)$	Gaussian distributed one-dimensional random variable with expectation μ and variance σ
$G_i(\mu, \sigma)$	Gaussian variable sampled anew for each possible value of the counter i

List of Acronyms

BB: Building Block

BBH: Building Block Hypothesis

EP: Evolutionary Programming

EPP: Evolutionary Progress Principle

ES: Evolutionary Strategy

GA: Genetic Algorithm

GRASP: Greedy Randomised Adaptive Search Procedures

GRH: Genetic Repair Hypothesis

FL: Fitness Landscape

ILS: Iterated Local Search

IPD: Ideal Probability Distribution

PD: Probability Distribution

PDF: Probability Density Function

PFL: Probable Fitness Landscape

QM: Qualitative Model

SA: Simulated Annealing

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Background

Evolutionary Algorithms (EAs) are stochastic, population-based, metaheuristic methods used to approximately solve optimisation problems. The popularity of EAs is due to their simplicity, flexibility, adaptability and robustness; qualities which make them ideal black-box methods[1]. These characteristics stem from the iterative process of Darwinian evolution inherent in these algorithms.

An English naturalist by the name of Charles Darwin published a book in 1859 entitled *On the Origin of Species* [42] in which he described the theory of evolution due to natural selection. In nature the *fitness* of an individual is evaluated according to its probability of survival. The fittest individuals in the population are *selected* by *survival of the fittest*, with the unfit individuals being eliminated from the population. The selected (non-eliminated) individuals then *reproduce* to create the next generation and the process repeats, as illustrated in Figure 1.1.

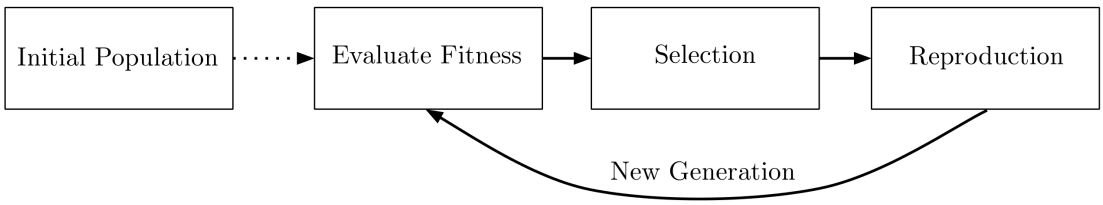


FIGURE 1.1: *Iterative process of Darwinian evolution.*

This same process occurs in EAs. Consider a function $f : \mathcal{S} \mapsto \mathbb{R}$ defined by an optimisation

[1] A *black-box* method is defined as method which is generally effective for input-output processes where the actual process is opaque (black), that is, only the inputs and their outputs may be known.

problem which is to be maximised[2]. The function f is known as the *fitness function* and $\mathcal{S}$ is the *search space*. The process of evolution begins with an initial population consisting of randomly generated *individual candidate solutions* from the search space. The *fitness* of an individual $s \in \mathcal{S}$ is *evaluated* according to its fitness value $f(s)$. The fittest individuals in the population are *selected*, then *reproduce* to create the next generation and the process repeats.

Underlying the process of evolution are the notions of *inheritance*, *selection* and *variation*[3]. Inheritance ensures that individuals will reproduce to create new individuals of similar fitness. Since only the fittest individuals are selected to produce the next generation, individuals in the new generation should be fitter than individuals in the previous generation. Hence the fitness of the population is expected to increase over the generations.

Variation is introduced into the population via new individuals having *similar*, but not the exact same, fitness as the individuals from which they were produced. This is balanced by selection which, through eliminating unfit individuals, removes variation from the population. Without variation, selection would cause the population to converge to multiple copies of the fittest individual that was present in the initial population. Having variation enables new (potentially fitter) individuals to be created and the fitness of the population not to be bound by the initial population.

The combination of inheritance and selection (increasing the fitness of the population over the generations) with variation (allowing new individuals to enter the population) results in the increase of the fitness of the population bounded only by the maximum of the fitness function. The individual candidate solutions are expected to become good approximations of the optimum solution as their fitness values increase toward the maximum and thus the EA is expected to provide good solutions to the optimisation problem.

EAs often determine the exact optimum of the fitness function, although there is no guarantee of convergence (at least for a finite population size or finite number of generations). This is due to the stochastic nature of EAs, which can cause different runs of the same EA with respect to the same problem to have completely different results.

It is expected that for a particular problem an EA will, on average, have worse performance than an algorithm designed for that problem. This is a consequence of the No Free Lunch theorems [52, 180], which state that there is no universal algorithm which is appropriate for all optimisation problems. Instead, algorithms are appropriate to a *problem class* — algorithms which exploit properties of a problem class will have superior performance in that class. EAs have, however, proven to be rather successful black-box optimisation algorithms [180], *i.e.* they are successful in large problem classes modelling a variety of real-world problems [29, 34, 88, 152, 190]. In fact, recently a new empirical methodology has managed to show that certain metaheuristics do have superior performance in the problem class of binary real-world problems [62]. However, due to their generality, they do not exploit properties of any particular problem and will usually be outperformed by an algorithm specifically designed to exploit a problem's properties.

Another consequence of the No Free Lunch theorems is that all theoretical and empirical results are specific to a problem class. This, combined with the stochastic nature of EAs, make EAs difficult to research.

EA theory has been an intense area of research over the last few decades, producing many

[2] Any optimisation problem can be transformed to a maximisation problem by considering the negative of the objective function, since $\min[f(x)] = -\max[-f(x)]$.

[3] These are the essential properties of Universal Darwinism. Any system with these properties should exhibit Darwinian evolution [27, 44, 103].

achievements, such as the *schema theorem* and *building block hypothesis* [68, 83, 122], convergence theorems [4, 146, 148, 164] and time-complexity runtime analyses [4, 33, 101, 146]. Despite the vast body of research on EAs, there is no established mathematical theory explaining *why* EAs work and the reason for their success. Papadimitriou and Steiglitz [134] affirmed this claim as follows: "Although very little has been rigorously established about the performance of such algorithms, they often seem to do remarkably well on certain problems. Developing the mathematical methodology for explaining and predicting the performance of these and other heuristics is one of the most important challenges facing the fields of optimization and algorithms today." Mühlenbein [124] even expressed doubt as to the feasibility of establishing a universal theory for EAs: "Given the mathematical difficulty of the infinite population size model, we doubt that a mathematical analysis of finite populations will be possible."

As a result, empirical simulations of problem instances are often used to demonstrate the performance of algorithms in various problem classes [89]. The disadvantage of this approach is that many problem instances need be tested in order to accurately represent a problem class. Even then this does not guarantee performance, nor does it yield explicit explanations that build intuition, essential for the design of new algorithms. As Cohen [38] puts it (according to [175]): "It is good to demonstrate performance, but it is even better to explain performance."

Explanations are often attempted using terms such as *exploitation*, *exploration*, *intensity* and *diversity*; but there are no universally agreed upon definitions for these terms [54, 128]. This limits their explanatory powers, unless specific definitions are given in each context of use. Even so, the lack of universal definitions limits the effectiveness of these terms in a general context.

Goldberg, in his book *The Design of Innovation* [69], outlines a spectrum of approaches for analysing and modelling algorithms. The spectrum stretches from rough intuitive models on the far left, to exact technical models on the far right, as shown in Figure 1.2.

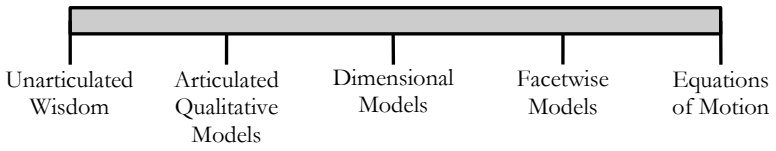


FIGURE 1.2: *The modelling spectrum of Goldberg [69].*

The *equations of motion* refer to the equations that govern the algorithm. Solving these equations is the ultimate goal, as such a model describes the performance of an algorithm exactly. If this is not achievable (or practical), then Goldberg advocates the use of simplified *facetwise models*, which analyse a certain facet of the performance of an algorithm. These facets can be patch-quilt integrated into more extensive *dimensional models*. The simplest type of models are articulated *Qualitative Models (QMs)*, which may range from "verbal descriptions of mechanism to pictorial or graphical representations of processes or relationships"[69]. In the absence of any models, *unarticulated wisdom* prevails.

As of yet, there is no soluble set of equations of motion for EAs. There are many facetwise and dimensional models that describe certain aspects of EAs, but they are too patchy to form a complete understanding of EAs. Unarticulated wisdom, although perhaps useful for an individual, is not *research* since it is not communicated to the research community. Hence QMs are appealed to, in order to complement and patch together other models through explanations. Unfortunately these appeals usually fail due to inarticulate terminology [54] and the lack of a widely accepted formal QM.

1.2 Problem statement

The concern of this thesis is a universal fundamental analysis of EAs. The term *universal* is essential. It implies that the analysis is not particular to a specific problem or small problem class. A direct consequence is that it cannot involve analysing the performance of EAs, since according to the No Free Lunch theorems this will simply be equal to the average. Instead it can only entail an analysis of the fundamental principles of EAs.

A QM is the appropriate approach for analysing the fundamental principles of EAs, as it does not involve analysis of performance. Instead it traces the historical development of EAs, identifies the principles of EAs and then deduces the consequences of these principles. This offers insight into why EAs work as well as provides a platform for analysing EA operators.

The aim of this thesis is therefore to establish a QM of EAs. Its scope is limited to the near universal problem class of continuous functions on a continuous search space. This problem class is of great relevance as many practical problems are of this form and much work involving EAs has been devoted to solving problems in this class. Since all metaheuristics depend on some notion of continuity [62, p.2129], the scope is barely limited by the criterion of continuity.

The QM is required to explain the behaviour of certain EAs. Specifically, this is done for the three dominant algorithms: *Genetic Algorithms* (GAs), *Evolution Strategies* (ESs) and *Evolutionary Programming* (EP). Each of these algorithms have unique selection and reproduction operators, which merit their own analyse.

1.3 Objectives and thesis organisation

This thesis is organised into seven chapters, each of which focus on a specific objective pursued in this thesis. Each chapter is listed below with a brief description of its content:

1. **Introduction**
   - To *introduce* EAs
2. **The Development of Evolutionary Algorithms**
   - To *trace* the historical development of EAs
3. **Principal Operators of Evolutionary Algorithms**
   - To *document* the prevalent mutation and selection EA operators
4. **The Probable Fitness Landscape**
   - To *propose* the notion of a *Probable Fitness Landscape* (PFL) as the basis of a QM of EAs on continuous search spaces
   - To *deduce* the consequences of the PFL model with regard to various terminologies
   - To *extend* the PFL model to the *Ideal Probability Distribution* (IPD), which may be used to analyse the performance of operators
5. **Genetic Algorithms**
   - To *present* and *explain* the performance of GAs

6. **Evolution Strategies and Evolutionary Programming**
    - To *present* and *explain* the performance of ESs and EP
7. **Conclusion**
    - To *defend* the use of QMs
    - To *conclude* the thesis with suggestions for *further work*.

Chapters 1–3 aim to introduce the main parts and principles behind EAs, which set the foundation for novel contributions in Chapters 4–6. The final concluding chapter coalesces the analysis of the previous chapters and proposes topics for future work.

CHAPTER 2

The Development of Evolutionary Algorithms

Contents

Consider a continuous fitness function $f : \mathcal{S} \mapsto \mathbb{R}$, where $\mathcal{S} \subset \mathbb{R}^n$ is a Lebesgue measurable search space (also known as the *parameter space*). The objective is to find the global maxima[1] of f, although points close to a global maximum may be sufficient for practical purposes.

Metaheuristic methods (also known as *metaheuristic algorithms* or simply *metaheuristics*) is a modern type of computational optimisation algorithm for determining such points. Unlike most classical optimisation algorithms (*e.g.* the Newton-Raphson method, conjugate gradient method or simplex method), metaheuristics do not use gradients nor are they deterministic [23]. These qualities make metaheuristics ideal black-box methods [180]. Typically, metaheuristics do not guarantee convergence to a point of global maximum, although they usually produce good solutions in a reasonable amount of time.

The process of metaheuristics works as follows. *Candidate solutions* are iteratively generated in the search space and their *fitness* (the value of the fitness function corresponding to the points of the candidate solutions) is evaluated. The information from previous candidate solutions may be used to generate new candidate solutions of potentially higher fitness. Over the iterations the candidate solutions are expected to converge toward a point of global maximum and at the

[1] A global maximum is a value $\hat{s} \in \mathcal{S}$ for which $f(\hat{s}) \geq f(s)$ for all $s \in \mathcal{S}$. Whereas a global maximum is the maximum over the entire search space, a local maximum is the maximum in a local neighbourhood of the search space. A local maximum is a value $s' \in \mathcal{S}$ for which there exists an $\epsilon > 0$ such that $f(s') \geq f(s)$ for all $s \in \mathcal{S}$ satisfying $||s' - s|| < \epsilon$.

end of the run the candidate solution with the highest fitness is returned as the approximate (perhaps exact) maximum.

The first metaheuristic was probably used by Alan Turing during the Second World War[2] [39] (according to [187]). Since then they have undergone many stages of development, each addressing a limitation of previous methods. The major advances of metaheuristics, from pure random search to EAs, are outlined in this chapter.

2.1 Pure random search

Introduced by Brooks [25] in the 1950s, *pure random search* (also known as *uniform search* or *blind search*) is one of the first, and simplest, metaheuristics. The algorithm involves generating independent candidate solutions from the search space, with each point in the search space typically having an equal probability of being generated. The pure random search method is given in pseudocode form as Algorithm 2.1.

Algorithm 2.1: Pure random search

Input : Fitness function f, search space $\mathcal{S}$ and number of iterations N
Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
s* ← randomly generated point in S;
for k ← 2 to N do
    s ← randomly generated point in S;
    if f(s) > f(s*) then
        s* ← s;
    end
end
```

It can be shown that the candidate solutions of the algorithm converges toward the global optimum. Let s^k be the value of s^* at the end of the k^{th} iteration. Then the following theorem holds [193] (according to [6][3]).

Theorem 2.1.1. *The sequence $s^1, s^2, \ldots$ of random vectors generated by Algorithm 2.1 converges in probability to a global maximum $\hat{s}$.*

Proof. Let $U_\epsilon(\hat{s}) \equiv \{s \in \mathcal{S} : |f(s) - f(\hat{s})| < \epsilon\}$ and μ be the standard measure in $\mathbb{R}^n$. Then, for arbitrary $\epsilon > 0$, the probability that the k^{th} vector in the sequence is contained in U_ϵ is

$$P\{s^k \in U_\epsilon(\hat{s})\} = 1 - \left(1 - \frac{\mu(U_\epsilon(\hat{s}))}{\mu(\mathcal{S})}\right)^k \xrightarrow{k\to\infty} 1.$$

Therefore the sequence converges in probability, phrased mathematically, as

$$\lim_{k\to\infty} P\{s^k \notin U_\epsilon\} = 0$$

for all $\epsilon > 0$. □

[2] Turing referred to his algorithm as a *heuristic search*, since it only worked most of the time. The term *metaheuristic* is a combination of the term *meta-*, meaning "higher level", and *heuristic*, meaning "to find". Metaheuristics may be viewed as heuristic methods which are general and are easily modified to solve a range of problems.

[3] The theorem presented here is different to that referenced in [6], as there it is stated that *convergence with probability one* holds yet only *convergence in probability* is proven.

To give a concrete example of the convergence of Algorithm 2.1 (taken from [6]), assume $\mathcal{S}$ to be a hypersphere in $\mathbb{R}^n$ of radius R. The measure of $\mathcal{S}$ is

$$\mu(\mathcal{S}) = \frac{R^n \pi^{n/2}}{\Gamma(\pi/2 + 1)}$$

and likewise

$$\mu(U_\epsilon(\hat{s})) = \frac{\epsilon^n \pi^{n/2}}{\Gamma(\pi/2 + 1)},$$

where Γ denotes the Gamma function. Their volume ratio is

$$\frac{\mu(U_\epsilon(\hat{s}))}{\mu(\mathcal{S})} = \left(\frac{\epsilon}{R}\right)^n.$$

Thus, in order to reach at least a probability p^* for $s^k \in U_\epsilon(\hat{s})$,

$$p^* = 1 - \left(1 - \left(\frac{\epsilon}{R}\right)^n\right)^k,$$

or solving for the number of trials,

$$k = \frac{\ln(1 - p^*)}{\ln(1 - \left(\frac{\epsilon}{R}\right)^n)} \approx -\ln(1 - p^*)\left(\frac{\epsilon}{R}\right)^n,$$

using the approximation $\ln(1 + x) \approx x$ for $x \ll 1$. This demonstrates the exponential growth of computation time depending on n. In fact, for all Lipschitz continuous fitness functions, the expected number of iterations of a pure random search is proportional to the exponential of n [192] (according to [191]).

As an example of how pure random search works, consider the fitness function $f(s) = -s^2$ and the search space $[-1, 1]$. A run of ten iterations of the pure random search algorithm may give the results displayed in Table 2.1 and Figure 2.1.

It can be seen that the maximum fitness is generated during iteration 4 with the candidate solution of $s = -0.03$, corresponding to a fitness of -0.01. This is quite close to the maximum point of $s = 0$, corresponding to a fitness of 0, hence the algorithm has produced a good solution.

After the algorithm found a near maximum point at iteration 4 it did not manage to improve on it. In fact, the candidate solution generated during iteration 10 has the second worst fitness. There is no sense that the candidate solutions were converging toward the maximum, that better candidate solutions were being generated using the knowledge gained during previous iterations. This is due to the constant probability distribution (the uniform distribution across the entire search space) being used to generate candidate solutions, evident on line 3 of Algorithm 2.1.

A *global random search*, as defined by Zhigljavksy [193] (according to [6]), allows for the construction of a new probability distribution at each iteration. By incorporating information from previous candidate solutions in constructing the probability distribution, it may be adapted in order to generate better candidate solutions. All of the following algorithms in this section take advantage of this observation and are examples of global random search.

Iteration (i)	Candidate solution (s)	Fitness ($f(s)$)
1	-0.69	-0.48
2	0.94	-0.89
3	0.91	-0.83
4	$\mathbf{-0.03}$	$\mathbf{-0.01}$
5	0.60	-0.36
6	-0.72	-0.52
7	-0.16	-0.03
8	0.83	-0.69
9	0.50	-0.25
10	0.91	-0.83

TABLE 2.1: *Fitness function with values of candidate solutions generated by a pure random search for the maximum of* $f(s) = -s^2$ *in the search space* $s \in [-1, 1]$.

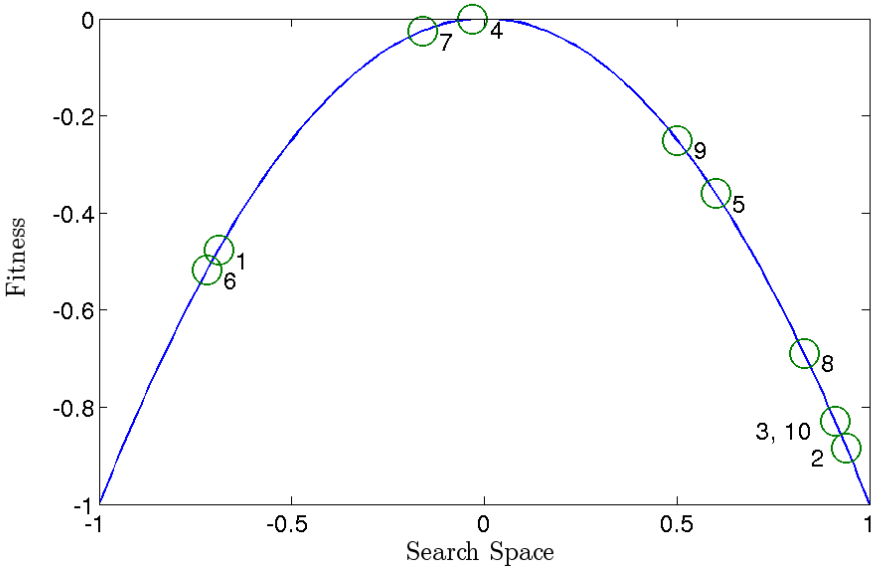


FIGURE 2.1: *Plot of the fitness function with the candidate solutions generated by a pure random search for the maximum of* $f(s) = -s^2$ *in the search space* $s \in [-1, 1]$.

2.2 Hill climbing

Hill climbing is the simplest process whereby an algorithm can learn from the information gained from previous iterations to generate better candidate solutions. Instead of generating candidate solutions from the uniform probability distribution of the entire search space, as is the case in pure random search, hill climbing only generates candidate solutions in the local neighbourhood of the current best candidate solution. This ensures that the search focuses on the region of the search space around good solutions as, due to the continuity of the fitness function, other solutions of high fitness are likely to be found in this region. Through incremental local improvements, the candidate solutions should move in the direction of positive gradient and thereby "climb the hill" to converge toward a local maximum of the fitness function.

The hill climbing method is given in pseudocode form as Algorithm 2.2. It differs from Algorithm 2.1 only on line 3 where the probability distribution of the local neighbourhood is specified instead of the entire search space.

Algorithm 2.2: Simple hill climbing

Input : Fitness function f, search space $\mathcal{S}$ and number of iterations N
Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
1 s* ← randomly generated point in S;
2 for k ← 2 to N do
3 |   s ← randomly generated point in the local neighbourhood of s*;
4 |   if f(s) > f(s*) then
5 |   |   s* ← s;
6 |   end
7 end
```

Again consider the example of the fitness function $f(s) = -s^2$ and the search space $[-1, 1]$. For a hill climbing algorithm the local neighbourhood around any point $s \in \mathcal{S}$ must be defined and in this case the interval $[-0.3 + s, 0.3 + s]$ is used (this is one of the simplest and oldest techniques, as discussed in the 1963 paper by Karnopp [94]). A run of ten iterations of Algorithm 2.2 may give the results displayed in Table 2.2 and Figure 2.2.

It is evident from the table and figure that the candidate solutions' fitness did improve over the iterations, climbing up the hill, with the final iteration giving the best candidate solution. This is an advance on pure random search, for which the candidate solutions did not exhibit any iterative improvement toward the maximum.

A characteristic of *simple* hill climbing is that points are only generated in the *local* neighbourhood of the current best candidate solution; hence it is an example of a *local search*. Simple hill climbing excels at finding local maxima, yet may not be able to find a global maximum, which is ultimately what is of interest [13]. This is because the algorithm may "get stuck on the top of a hill" which is not a global maximum — a problem known as *premature convergence*. Technically, premature convergence is the state in which the candidate solution with the highest fitness previously generated is suboptimal and no candidate solutions with a higher fitness can be generated [102]. Due to this phenomenon, simple hill climbing does not guarantee convergence toward the global maximum and, therefore, there are no expressions for the expected number of iterations or convergence probability.

Iteration	Candidate solution	Fitness
1	-0.90	-0.81
2	-0.75	-0.56
3	-0.67	-0.45
4	-0.92	-0.84
5	-0.93	-0.86
6	-0.51	-0.26
7	-0.26	-0.07
8	-0.24	-0.06
9	-0.48	-0.23
10	$\mathbf{-0.05}$	$\mathbf{-0.01}$

TABLE 2.2: *Fitness function with values of candidate solutions generated by hill climbing for the maximum of* $f(s) = -s^2$ *in the search space* $s \in [-1, 1]$.

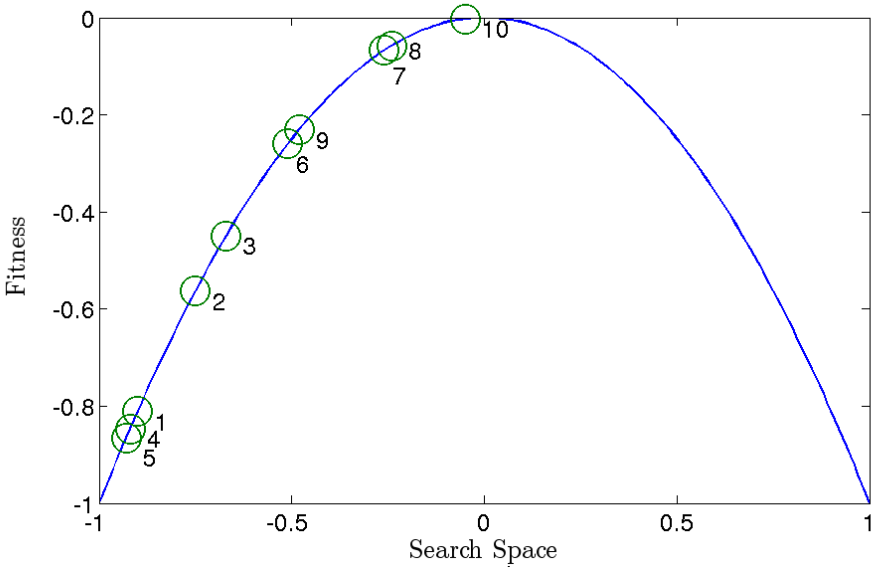


FIGURE 2.2: *Plot of a fitness function with the candidate solutions generated by hill climbing for the maximum of* $f(s) = -s^2$ *in the search space* $s \in [-1, 1]$.

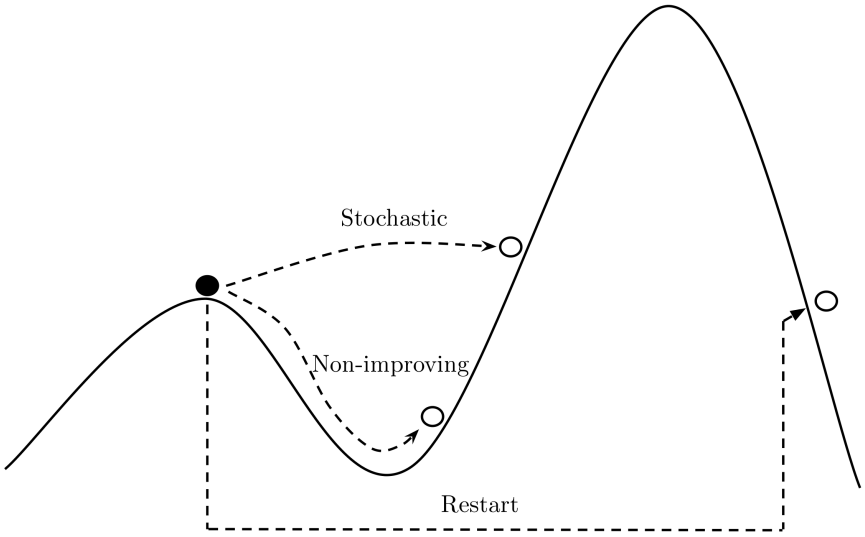


FIGURE 2.3: *Plot of the fitness function with the current candidate solution at a local maximum. The three escape strategies* restarting local search, stochastic search *and* accepting non-improving solutions *are displayed.*

In order to deal with premature convergence, an algorithm must have the ability to escape the neighbourhood of a local non-global maximum and move toward the region of a global maximum. There are three prevalent escape strategies: *restarting local search, stochastic search* and *accepting non-improving solutions*. These strategies are displayed schematically in Figure 2.3 and are discussed in the following sections.

2.3 Restarting local search — GRASP and ILS

The simplest method of escaping to a new region in the search space, and thereby avoiding premature convergence, is *random restart* [108]. It achieves this by generating the next candidate solution at a random point (with uniform probability, like in pure random search), from which the search continues. This technique, although not practically efficient, is revealing in how it explicitly alternates between hill climbing and pure random search. It has been argued that all metaheuristics are ultimately elaborate combinations of hill climbing and pure random search [58, 109].

Random restart is the most basic member of a family of algorithms known as *Greedy Randomised Adaptive Search Procedures (GRASP)* [56, 57, 85, 142]. GRASP use a *construction operator* to generate restart points such that they are diverse and close to a maximum, and therefore suitable for a subsequent local search. Typically, GRASP do not utilise the information of previous candidate solutions (GRASP-like algorithms which do use this information are known as *Adaptive Iterated Construction Search* [85]). The GRASP method is described in pseudocode in Algorithm 2.3.

Iterated Local Search (ILS) [26, 85, 107, 108, 166] generates restart points by *perturbing* the position of a candidate solution. These perturbations effectively generate new candidate solutions in a sub-search space around a previously generated candidate solution, instead of in the

Algorithm 2.3: GRASP

Input : Fitness function f, search space $\mathcal{S}$, number of iterations N, construction operator `Construction`$(\cdot)$ and local search operator `LocalSearch`$(\cdot)$

Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
s* ← randomly generated point in S;
for k ← 2 to N do
    s ← Constrution(S);
    s' ← LocalSearch(s);
    if f(s') > f(s*) then
        s* ← s';
    end
end
```

entire search space as is the case for GRASP. If the strength of the perturbation is appropriate then new candidate solutions should be generated outside of the neighbourhood of the local maximum, yet still in a region with points of high fitness. ILS requires a *selection operator* (also known as an *acceptance criterion*) to decide whether to perturb around the current or previous candidate solution. The selection operator may be deterministic (always selecting the candidate solution with the higher fitness) or stochastic. The ILS method is described in pseudocode in Algorithm 2.4.

Algorithm 2.4: ILS

Input : Fitness function f, search space $\mathcal{S}$, number of iterations N, perturbation operator `Perturbation`$(\cdot)$, local search operator `LocalSearch`$(\cdot)$ and selection operator `Select`$(\cdot,\cdot)$

Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
s* ← randomly generated point in S;
s'' ← s*;
for k ← 2 to N do
    s ← Perturbation(s'');
    s' ← LocalSearch(s);
    s'' ← Select(s'', s');
    if f(s') > f(s*) then
        s* ← s';
    end
end
```

GRASP and ILS belong to the set of metaheuristics known as *two-phase methods* [144, 153] or *multi-start methods* [115]. These methods consist of a global phase, which generates *feasible points*, coupled with a local phase, capable of finding a local maximum. An ever-open question in designing effective two-phase methods is how often the global phase should be employed, or equivalently, for how long each local phase should run [114]? Every time a local phase is terminated, the region around a local (potentially global) maximum is left. If terminated prematurely, the local phase would not have had enough time to find the maximum and the computations performed in generating candidate solutions close to the maximum are wasted. On the other hand, if the local search is run for too long, then computational time is wasted on generating candidate solutions in regions of low fitness, or trying to find points of higher fitness

when the algorithm has already converged.

2.4 Stochastic search

An alternative to two-phase methods is *stochastic search* (also known as *random search* or *random optimisation*). Stochastic search differs from local search in one crucial regard: in local search, new candidate solutions are *always* generated in the neighbourhood of the current candidate solution, whereas in stochastic search this is not necessarily, but only *probably*, the case [176]. Thus, stochastic search is both local and global search, and only one phase is necessary.

Around the 1960s stochastic search was developed by the pioneering Anderson [3], Rastrigin [125, 139], Matyas [117, 118], Schumer [155] and Steiglitz [155]. Like local search, stochastic search is a sequential optimisation algorithm (also known as a *trajectory method*) for which the progression of the candidate solution s may be represented by

$$\begin{aligned} s^{k+1} &= s^k + \Delta s^k, \\ \Delta s^k &= \sigma^k d^k, \qquad k = 0, 1, \ldots, \end{aligned}$$

where s, Δs and d are N-vectors with d normalised ($||d|| = 1$), σ is a scalar and superscripts denote iteration numbers [154]. The Δs term is called the *step*, with d and σ referring to the *step direction* and *step size*, respectively. There are two main concerns when constructing such a method: the direction problem, that is to determine d, and the step size problem, that is to determine σ. In most random searches the direction d is chosen using random vectors (an exception being the bias vectored algorithm of Matyas, discussed in [23]), whereas the step size is more tightly controlled.

In the following subsections it is demonstrated that stochastic search is faster than pure random search and that it too can have guaranteed convergence. Also, the notions of optimum step size and adaptable step size are considered.

2.4.1 Fixed step size random search

Rastrigin [125, 139] proposed a primitive stochastic search, christened *fixed step size random search*, for which the step direction is random, but the step size is constant. The search is worthwhile examining for both its historical and analytical value, as it was used in the first mathematical description of the rate of convergence of a stochastic search. Thus, it laid the mathematical foundation for the field of metaheuristics.

Consider fixed step size random search "with reversing," which can be described as follows. Let the search be carried out in steps of unit length, *i.e.* $\sigma = 1$, and the direction vector d be chosen at random, with equal probability for all directions. If, as the result of a random step, the fitness increases, then the step is accepted, whereas, if the fitness is not increased, then the step is rejected. The algorithm is described in pseudocode as Algorithm 2.5. The following result shows that fixed step size random search is faster than pure random search (at least for high-dimensional fitness functions).

Theorem 2.4.1 ([139])**.** *The rate of convergence of Algorithm 2.5 in the case of a linearised fitness function increases proportional to the square root of the number of degrees of freedom.*

Proof. Consider a point in the search space away from the extremum of the function (where the gradient is non-zero) in which case the function can be expanded as a Taylor series keeping

Algorithm 2.5: Fixed step size random search

Input : Fitness function f, search space $\mathcal{S}$ and number of iterations N
Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
s* ← randomly generated point in S;
s^1 ← s*;
for k ← 1 to N do
    Δs^k ← randomly generated vector of unit length;
    if f(s^k + Δs^k) > f(s^k) then
        y^k ← 1;
        else y^k ← 0;
    end
    s^{k+1} ← s^k + y^k Δs^k;
    if f(s^{k+1}) > f(s*) then
        s* ← s^{k+1};
    end
end
```

only the linear terms. The coordinates of the search space may be rotated such that the direction of the gradient is $(x_1, x_2, \ldots, x_n) = (1, 0, \ldots, 0)$. Let the coordinates of the step vector $\Delta s = (\Delta s_1, \ldots, \Delta s_n)$ be transformed into spherical coordinates, that is

$$\begin{aligned}
\Delta s_1 &= r\cos\theta_1 \\
\Delta s_2 &= r\sin\theta_1\cos\theta_2 \\
&\vdots \\
\Delta s_{n-1} &= r\sin\theta_1\sin\theta_2\ldots\sin\theta_{n-2}\cos\phi \\
\Delta s_n &= r\sin\theta_1\sin\theta_2\ldots\sin\theta_{n-2}\sin\phi,
\end{aligned}$$

where $r \in [0, \infty)$, $\theta_i \in [0, \pi]$ and $\phi \in [0, 2\pi)$. In the search space, consider the plane passing through the step vector and the direction of the gradient, displayed in Figure 2.4. Steps with angles θ_1 in the range $[0, \pi/2)$ will yield an increase in the fitness, while steps with θ_1 in the range $[\pi/2, \pi]$ will not. The probability of each of the two types of step is $1/2$.

The mean displacement in the direction of the gradient for one successful step, *i.e.* for $0 \leq \theta_1 < \pi/2$, is the integral of inner product of the step vector and the gradient over the surface of possible step vectors, divided by the total surface area of possible step vectors. Since $\sigma = 1$, the surface area is that of the unit hypersphere. The area element, that is the determinant of the Jacobian of the transformation of Cartesian coordinates to spherical coordinates, is $|J| = r^{n-1}\sin^{n-2}\theta_1\sin^{n-3}\theta_2\ldots\sin\theta_{n-3}$ and the inner product of the step vector and gradient is $r\cos\theta_1$. Setting $r = 1$, the mean displacement in the direction of the gradient for one successful step as a function of the number of degrees of freedom is

$$\begin{aligned}
U(n) &= \frac{\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi}\int_0^{\pi/2}(\cos\theta_1)\left(\sin^{n-2}\theta_1\sin^{n-3}\theta_2\ldots\sin\theta_{n-3}\right)d\theta_1 d\theta_2\ldots d\theta_{n-2}d\phi}{\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi}\int_0^{\pi/2}\left(\sin^{n-2}\theta_1\sin^{n-3}\theta_2\ldots\sin\theta_{n-3}\right)d\theta_1 d\theta_2\ldots d\theta_{n-2}d\phi} \\
&= \frac{\int_0^{\pi/2}\cos\theta_1\sin^{n-2}\theta_1 d\theta_1}{\int_0^{\pi/2}\sin^{n-2}\theta_1 d\theta_1} \\
&= \frac{\Gamma(n-1)}{2^{n-3}(n-1)\left[\Gamma\left(\frac{n-1}{2}\right)\right]^2},
\end{aligned}$$

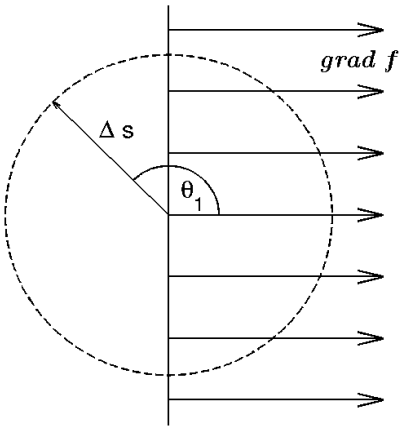


FIGURE 2.4: *A section of the search space for a linear fitness function in the proof of Theorem 2.4.1.*

where Γ is the gamma function.

If the probability of a successful and unsuccessful step is the same then, on average, there is one unsuccessful step for one successful step. Thus, the mean displacement for one successful step is reduced by half; and the mean displacement of the algorithm in the direction of the gradient per step of the search (*i.e.* the rate of convergence) is $U(n)/2$.

The *search loss* is defined as the number of steps in the system such that the projection of the vector sum of these steps on the direction of the gradient has the same length as one operating step, *i.e.* one. This is proportional to the expected number of steps required for convergence. For the above algorithm the mean search loss is

$$\begin{aligned} K_n &= 2/U(n) \\ &= 2\frac{2^{n-3}(n-1)\left[\Gamma\left(\frac{n-1}{2}\right)\right]^2}{\Gamma(n-1)} \\ &= \frac{2^{n-2}(n-1)\left[\Gamma\left(\frac{n-1}{2}\right)\right]^2}{\Gamma\left(\frac{n-1}{2}\right)\Gamma\left(\frac{n}{2}\right)2^{n-2}/\sqrt{\pi}} \\ &= \frac{\sqrt{\pi}(n-1)\Gamma\left(\frac{n-1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}, \end{aligned}$$

where the identity $\Gamma(x)\Gamma(x+1/2) = 2^{1-2x}\sqrt{\pi}\Gamma(2x)$ was used. Employing the inequalities

$$\frac{1}{\sqrt{x-\frac{1}{2}}} < \frac{\Gamma(x-\frac{1}{2})}{\Gamma(x)} < \frac{1}{\sqrt{x-\frac{3}{4}}}$$

it follows that, for $n > 2$,

$$\sqrt{2\pi}\sqrt{n-1} < K_n < \sqrt{2\pi}\sqrt{n}.$$

Hence the search loss for the fixed step size random search method, in the case of a linearised fitness function, increases proportional to the square root of the number of degrees of freedom. □

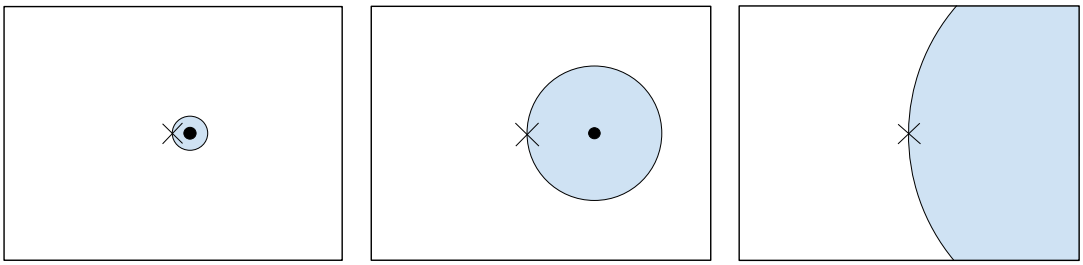


FIGURE 2.5: *Graphs representing regions used to generate the next candidate solution, corresponding to the step size, with (a) representing a large step size, (b) a moderate step size and (c) a small step size. The current candidate solution is indicated by a cross, the maximum by a black dot and the region of improvement is shaded.*

Rastrigin [139] goes on to discuss fitness functions that yield central fields in the search space, but the format of the argument is the same. It demonstrates that the fixed step size random search iteratively improves the fitness function, with the rate of convergence proportional to square root of the number of dimensions. This is considerably slower than the linear and exponential proportionality of gradient search and pure random search, respectively (as proven in Section 2.1), and confers a definite advantage on the stochastic search method (at least for fitness functions of a certain type).

2.4.2 Optimum step size

The limitations of fixed step size random search were recognised and discussed by Schumer and Steiglitz [155] who argued for an *optimum step size random search* (also discussed in [125]). A mathematical treatment for determining the theoretical optimum step size may be found in various papers [35, 154, 155]. The gist of the argument is summarized by Schumer and Steiglitz [155] as follows: "If the step size ... is very small, the probability of improvement is approximately one half, but the improvement is very small for a successful step, and this results in a small average improvement. On the other hand, if the step size is made too large, the step will overshoot the minimum and the probability of improvement will be extremely small, also resulting in a very small average improvement. Somewhere between these extremes lies an optimum step size, *i.e.* a step size for which the probability of the improvement of the quality function is not one half, but lies between zero and one half."

This concept is illustrated in Figure 2.5. In each diagram the current candidate solution s is indicated by a cross, the maximum by a black dot and the *region of improvement*, that is the set $\mathcal{S}_i \subset \mathcal{S}$ for which $f(s_i) > f(s)$ if $s_i \in \mathcal{S}_i$, is shaded. Figure 2.5(a) represents the entire search space, (b) a region contained in the search space centred around the current candidate solution and (c) an even smaller region contained in that of (b), also centred around the current candidate solution. The size of the *generating region* (the region in which the next candidate solution may be generated) is proportional to the step size, hence, Figure 2.5(a) corresponds to a large step size, (b) to a moderate step size and (c) to a small step size.

In Figure 2.5(a) the step size is large enough so that the generating region contains all points in the search space, including that of the maximum (as in pure random search). The advantage of a large step size is that the region of improvement is guaranteed to be in the generating region. This is negated by the generating region containing many points that are not in the region of improvement, resulting in a high probability of generating candidate solutions of a lower fitness

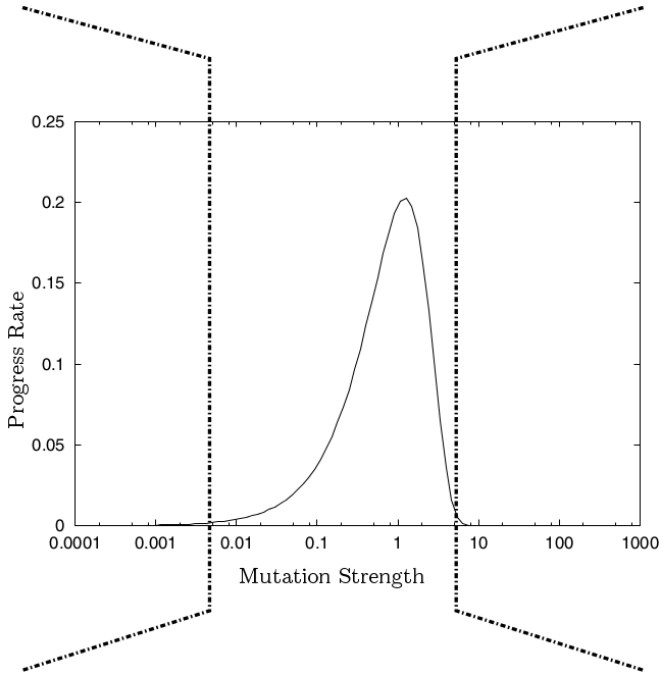


FIGURE 2.6: *The evolution window with the progress rate depending on the mutation strength [20].*

than that of the current candidate solution. In the limit as $\sigma \to \infty$, the probability of generating an individual outside of the region of improvement tends to one, and accordingly the progress rate[4] tends to zero.

This is contrasted to Figure 2.5(c) where the step size is so small that candidate solutions can only be generated close to the current candidate solution (as in simple hill climbing). In the extreme case as $\sigma \to 0$, the probability of generating an individual in the region of improvement becomes one half (as was discussed in Theorem 2.4.1). The disadvantage of a small step size is that the incremental improvement[5], $|\nabla f(s) \cdot d|\sigma$, is proportional to σ. Therefore, as $\sigma \to 0$ the incremental improvement tends to zero and likewise the progress rate. Even in the non-limit case, the generating region may only include a small subset of the region of improvement, possibly excluding the maximum (as is the case in Figure 2.5(c)).

In both the cases where $\sigma \to \infty$ or $\sigma \to 0$, the algorithm is incapable of improving the fitness of the candidate solutions and therefore the optimum step size must lie between these extremes. The range of suitable mutation strengths is represented by the *evolution window* (a term coined by Rechenberg [140] according to [20]), displayed in Figure 2.6. In order for an algorithm to have a high probability of success, the mutation strength (step size) must be chosen in the evolution window.

Typically, the optimum step size, or the shape of the evolution window, cannot be determined *a priori* and must be found with additional experimentation [155]. There are two techniques, which are often used in combination, for achieving an adequate step size: *adaptive step size* and *stochastic step size*.

[4]The *progress rate* refers to the number of iterations required until a global maximum is reached or, equivalently, the rate at which the distance between the candidate solutions and a global maximum decreases.

[5]Where ∇f is the gradient of f and d is the step direction vector.

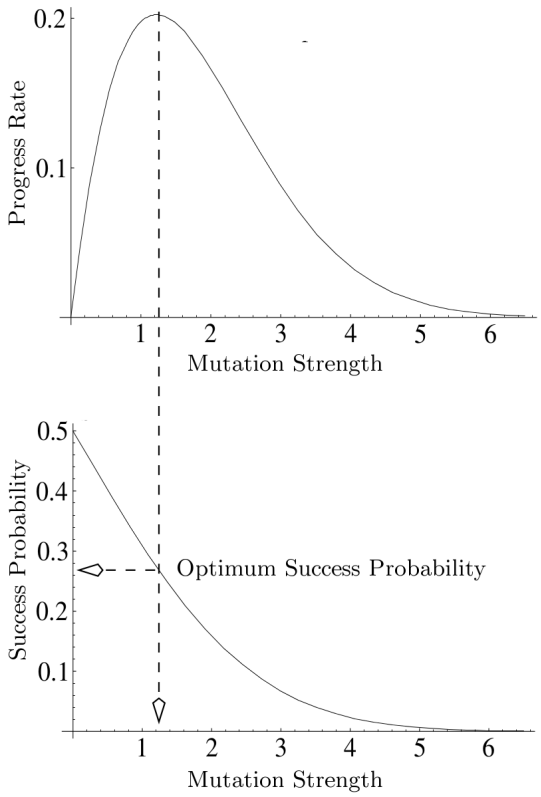


FIGURE 2.7: *The progress rate and probability of success as a function of the mutation strength for the sphere model [20]. The top figure plots the progress rate versus the step size, whereas the bottom figure plots the success probability versus the step size.*

2.4.3 The 1/5th-success rule and adaptive search

In Rechenberg's PhD thesis [140] (according to reference [20]) the optimum step size for the sphere model was examined. The fitness function of the sphere model is $f(s) = -||s||^2$. The normalised progress rate and success probability[6] for a stochastic search[7] was calculated (also done in reference [17, pp.64–69]) and is displayed graphically in Figure 2.7.

The optimum step size was determined to be $\sigma \approx 1.224$, which corresponds to a success probability of approximately 0.27. Although the optimum step size would have been difficult to approximate *a priori*, the success probability is expected to be roughly midway between 0 and 0.5. In fact, it is known from Section 2.4.2 that the progress rate is zero if the success probability is 0 or 0.5 (corresponding to a step size of 0 or ∞, respectively) and that it is non-negative for all success probabilities between 0 and 0.5. This suggests that the progress rate is a concave function of the success probability. The graph of the progress rate and success probability for the sphere model is displayed in Figure 2.8, which is clearly concave.

Rechenberg [140] went on to empirically investigate what success probability corresponds to the

[6]The *success probability* is the probability that the next candidate solution will be more fit than the current candidate solution.

[7]In this case the stochastic search known as (1+1)-ES was used (see Chapter 6), although any stochastic search would give similar results.

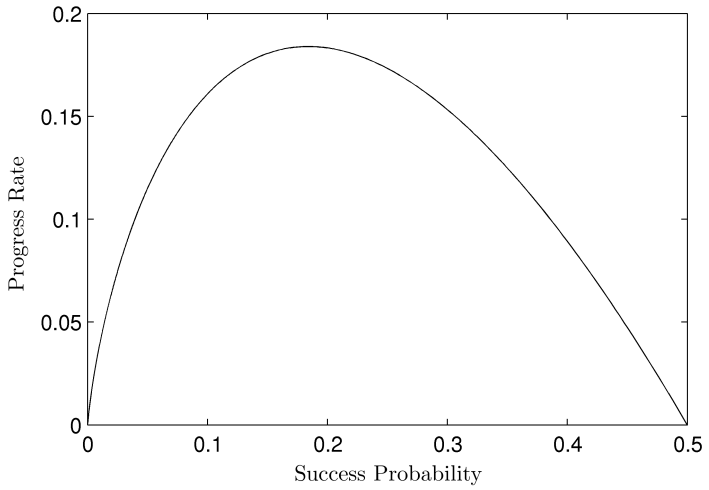


FIGURE 2.8: *The progress rate and probability of success (both depending on the mutation strength) for the sphere model [20].*

maximum progress rate. The totally different corridor function model rendered an optimum success probability of approximately 0.184, which is close to that of the sphere model (≈ 0.27). This led Rechenberg to conclude that for most optimisation problems the optimum success probability should be about 0.2.

In order to control the success probability, it is noted that the success probability is a strictly decreasing function of the step size. Thus, if the success probability should be increased, then the step size should be decreased, and vice versa. The 1/5th-success rule naturally follows, described in pseudocode form as in Algorithm 2.6.

Algorithm 2.6: 1/5th-success rule

Input : Number of iterations for which the step size is constant G and scaling factor $a < 1$

for *every* G *iterations* **do**
  G_s = number of successful candidate solutions during the previous G iterations
  $P_s = G_s/G$;
  $$\sigma = \begin{cases} \sigma/a & \text{if } P_s > 1/5 \\ \sigma \cdot a & \text{if } P_s < 1/5 \\ \sigma & \text{if } P_s = 1/5 \end{cases}$$
end

According to [6, 17], Schwefel [156] theoretically calculated the optimum value of the scaling factor to be $a \approx 0.817$, although practically recommended $a = 0.85$.

The 1/5th-success rule is an instance of an *adaptive search* (also known as *reactive search*). This type of search uses a rule based on the fitness of previous candidate solutions to adjust the step size (or any *strategy parameter*).

There are many types of adaptive search, including *Variable Neighborhood Search* [22, 77, 123, 168] and *Tabu Search* [22, 64, 65, 66, 168] (adapting the generating region) as well as *Guided Local Search* [22, 168, 173, 174] (adapting the fitness function). However, the most prevalent

adaptation technique is *σ-self-adaptation*, used in ESs and EP, which has been superseded by *cumulative step size adaptation* in *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES). These will be discussed in later chapters examining ESs and EP, respectively.

2.4.4 Matyas method

In 1963 Matyas [118] presented one of the first algorithms with both stochastic direction and stochastic step size. The algorithm is much like that of fixed step size random search (Algorithm 2.5), with differences in how the step vector is created and the criterion for acceptance. Instead of the step vector being of unit length, it is a Gaussian random vector with zero mean and unit correlation matrix. The *Probability Density Function (PDF)* $a(\cdot)$ of the step vector satisfies the following relation: for any finite, positive number c, there exists a $\theta > 0$ such that $a(s) > \theta$ for all $s \in \mathcal{S}$ satisfying $||s|| < c$. The acceptance criterion is stronger than that of fixed step size random search, as a step is only accepted if it improves the fitness by at least $\epsilon > 0$ (which is chosen *a priori*). The algorithm is described in pseudocode form as Algorithm 2.7.

Algorithm 2.7: Matyas method

Input : Fitness function f, search space $\mathcal{S}$, number of iterations N and PDF $a(\cdot)$
Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
s* ← randomly generated point in S;
s^1 ← s*;
for k ← 1 to N do
    Δs^k ← randomly generated vector sampled from a(·);
    if f(s^k + Δs^k) > f(s^k) + ε then
        y^k ← 1;
        else y^k ← 0;
    end
    s^{k+1} ← s^k + y^k Δs^k;
    if f(s^{k+1}) > f(s*) then
        s* ← s^{k+1};
    end
end
```

The original proof of convergence of Algorithm 2.7 was given by Matyas [118]. Two questionable points were identified by Baba [5], who formulated a better proof, presented below (another proof was given by Solis and Wets [160]).

Theorem 2.4.2 ([5])**.** *Let $\hat{s}$ be a point of maximum for f, that is to say*

$$f(\hat{s}) \geq f(s), \quad \text{for all } s \in \mathcal{S}. \tag{2.1}$$

Futhermore, let

$$R_\epsilon = \{s \in \mathcal{S} : |f(\hat{s}) - f(s)| < \epsilon\} \tag{2.2}$$

and assume that there exists some positive number r, such that

$$||s|| < r, \text{ for all } s \in \mathcal{S}. \tag{2.3}$$

Let the sequence of candidate solutions $\{s^k\}$ be created as in the method described above. Then $\{s^k\}$ converges with probability one to the region R_ϵ, that is[8]*:*

$$\lim_{k\to\infty} P\{\omega \in \Omega : s^k(\omega) \notin R_\epsilon\} = 0. \tag{2.4}$$

*Here, ω is the point of the probability measure space (Ω, B, P) (*i.e. *a random step vector), $s^k(\omega) = s^{k+1}$ with $\Delta s^k = \omega$, Ω is the basic ω-space (*i.e. *the space of all random step vectors), B is the smallest Borel field including $\cup_{k=1}^{\infty}\mathfrak{F}_k$ with $\mathfrak{F}_k = \sigma(\Delta s^1, \ldots, \Delta s^k)$ (*i.e. *the event space of random step vectors), and P is the probability measure.*

Proof. Since $f(s)$ is a continuous function, there exists a positive real number δ, such that

$$|f(\hat{s}) - f(s)| < \epsilon, \qquad \text{if } ||\hat{s} - s|| < \delta. \tag{2.5}$$

Consider the open sphere

$$A = \{s : ||\hat{s} - s|| < \delta\} \tag{2.6}$$

with centre $\hat{s}$ and radius δ. Clearly, from the relation (2.5) and (2.6),

$$R_\epsilon \supset A. \tag{2.7}$$

Assume that

$$s^k \in \mathcal{S}\backslash A, \quad k = 1, 2, 3, \ldots,$$

then the probability that s^{k+1} enters into the region A is

$$\begin{aligned} P_A\{s^{k+1}\} &= P\{s^{k+1} \in A : s^k \in \mathcal{S}\backslash A\} \\ &= P\{s^k + \Delta s^k \in A : s^k \in \mathcal{S}\backslash A\} \\ &= \int_A a(y - s^k)dy, \end{aligned} \tag{2.8}$$

where $a(\cdot)$ denotes the PDF of normal random vectors Δs^k. Let

$$\beta \equiv \inf_{\substack{y\in A \\ s^k\in\mathcal{S}\backslash A}} a(y - s^k).$$

From the assumption (2.3) that $||y - s^k|| < 2r$ and due to the properties of the PDF it may be concluded that

$$\beta > 0. \tag{2.9}$$

From (2.8) and (2.9),

$$P_A\{s^{k+1}\} \geq \beta\mu(A) \equiv \gamma, \qquad k = 1, 2, 3, \ldots, \tag{2.10}$$

where $\mu(A) = \int_A 1dy$ is the measure of A in $\mathbb{R}^n$.

Upon introduction of the random variables

$$y^i = \begin{cases} 1 & \text{if } f(s^i) \geq f(s^{i-1}) + \epsilon \quad \text{or} \quad s^i \in A \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 2, 3, 4, \ldots$$

[8]This is the criterion for *almost sure convergence* It is defined as follows: Let $\{s^k\}$ be a sequence of random vectors defined on a sample space Ω. The sequence $\{s^k\}$ is almost surely convergent to a region R_ϵ if there exists an event E such that $\{\omega \in \Omega : \{s^k(\omega)\}$ does not converge to $R_\epsilon\} \subseteq E$ and $P(E) = 0$.

and the following auxiliary variables

$$m \equiv \lfloor([f(\hat{s} - f(s^1))]/\epsilon\rfloor, \tag{2.11}$$

where $\lfloor x \rfloor$ is the largest integer which does not exceed x, it follows from the definition of the random variables y^k and (2.11) that s^k enters into the region R_ϵ if

$$\sum_{i=2}^{k} y^i \geq m + 1. \tag{2.12}$$

In view of (2.7) and (2.10), it follows that

$$\begin{aligned} P_M(x^{i-1}) &\equiv P\{y^i = 1 : s^{i-1} \in \mathcal{S}\backslash R_\epsilon\} \\ &\geq P\{s^{i-1} + \Delta s^{i-1} \in R_\epsilon : s^{i-1} \in \mathcal{S}\backslash R_\epsilon\} \\ &\geq P\{s^{i-1} + \Delta s^{i-1} \in A : s^{i-1} \in \mathcal{S}\backslash R_\epsilon\} \\ &\geq \gamma \end{aligned}$$

for all $i \geq 2$ and $s^{i-1} \in \mathcal{S}\backslash R_\epsilon$. Therefore, $1 \geq P_M(s^{i-1}) \geq \gamma$, and so

$$1 - \gamma \geq 1 - P_M(s^{i-1}) \geq 0. \tag{2.13}$$

Consider that $y^i = 1$ means a success and $y^i = 0$ means a failure. Then

$$\sum_{i=2}^{k} y^i = j$$

indicates a case in which j successes have occurred among $k-1$ trials. Therefore, it follows from (2.13) and Newton's theorem on the binomial probability distribution that

$$\begin{aligned} P\left\{\sum_{i=2}^{k} y^i = j : s^1, \ldots, s^{i-1} \in \mathcal{S}\backslash R_\epsilon\right\} &\leq \binom{k-1}{j}(1)^j(1-\gamma)^{(k-1)-j} \\ &= \binom{k-1}{j}(1-\gamma)^{(k-1)-j}. \end{aligned} \tag{2.14}$$

Hence,

$$P\{\omega \in \Omega : s^k(\omega) \notin R_\epsilon\} = P\left\{\sum_{i=2}^{k} y^i < m + 1 : s^1, \ldots, s^{i-1} \in \mathcal{S}\backslash R_\epsilon\right\} \tag{2.15}$$

$$= \sum_{j=0}^{m} P\left\{\sum_{i=2}^{k} y^i = j : s^1, \ldots, s^{i-1} \in \mathcal{S}\backslash R_\epsilon\right\} \tag{2.16}$$

$$\leq \sum_{j=0}^{m} \binom{k-1}{j}(1-\gamma)^{(k-1)-j}$$

by (2.12) and (2.14). Let G satisfy the inequality

$$G \geq (1-\gamma)^{-1-j}, \qquad \text{for all } j = 0, \ldots, m.$$

Furthermore, let $k > 2m$. Then,

$$\begin{aligned} P\{\omega \in \Omega : s^k(\omega) \notin R_\epsilon\} &\leq \sum_{j=0}^{m} \binom{k-1}{j} (1-\gamma)^k G \\ &\leq \sum_{j=0}^{m} \binom{k}{j} (1-\gamma)^k G \\ &\leq (m+1)\binom{k}{m}(1-\gamma)^k G \\ &\leq (m+1)k^m(1-\gamma)^k G/m!. \end{aligned}$$

Since

$$\lim_{k\to\infty} k^m(1-\gamma)^k = 0,$$

it follows that

$$\lim_{k\to\infty} P\{\omega \in \Omega : s^k(\omega) \notin R_\epsilon\} = 0,$$

as required. □

The above proof is based on that of Baba's [5], yet is not identical to Baba's proof; the author has found and rectified an error in [5]. In the original proof Baba introduces $\bar{\epsilon} = \epsilon/2$ and the region $K = \{s : |f(\hat{s}) - f(s)| \leq \bar{\epsilon}\}$ for determining $m \equiv \lfloor([f(\hat{s} - f(s^1))]/\bar{\epsilon}\rfloor$, as in (2.11). These notions are used in equation (23) (according to the numbering in Baba's paper), which states that

$$\begin{aligned} P\{\omega \in \Omega : s^k(\omega) \notin R_\epsilon\} &\leq P\left\{\omega \in \Omega : s^k(\omega) \notin R_\epsilon\right\} \\ &\leq P\left\{\sum_{i=2}^{k} y^i < m+1 : s^1, \ldots, s^{k-1} \in \mathcal{S}\backslash K\right\} \\ &= \sum_{j=0}^{m} P\left\{\sum_{i=2}^{k} y^i = j : s^1, \ldots, s^{k-1} \in \mathcal{S}\backslash K\right\} \qquad (2.17) \\ &\leq \sum_{j=0}^{m} P\left\{\sum_{i=2}^{k} y^i = j : s^1, \ldots, s^{k-1} \in \mathcal{S}\backslash R_\epsilon\right\}, \qquad (2.18) \end{aligned}$$

where (2.17) was added for clarity. Since $K \subset R_\epsilon$, (2.18) does not follow from (2.17); hence, the proof in [5] is erroneous. Fortunately, the proof can be repaired by completely excluding the notion of $\bar{\epsilon}$ and K, as is done in (2.11), (2.15) and (2.16).

An even simpler proof may be achieved by recognizing each iteration as a Bernoulli trial. The probability of successfully entering A each iteration is bounded below by γ, and thus the probability of not having entered A after k steps is less than $(1-\gamma)^k$, which goes zero as $k \to \infty$.

The above theorem demonstrates that convergence can be guaranteed for a search with stochastic step size. This enables the iterative improvement ability of hill climbing to be combined with the guaranteed convergence of pure random search. The convergence properties of the algorithm can be strengthened further by allowing non-improving solutions, as discussed in the following section.

2.5 Accepting non-improving solutions — Simulated Annealing

The third and final escape strategy from a local maximum is *accepting non-improving solutions*. It is related to the *Select* operator in ILS (given in Algorithm 2.4) which is used to select either the current or the previous candidate solution (from which the next candidate solution is to be generated). In all of the algorithms discussed up to this point (excluding ILS) the candidate solution with the greater fitness is always selected. Now the candidate solution with the greater value is only *probably* selected. This allows less fit (*non-improving*) candidate solutions to occasionally be selected, enabling the algorithm to "climb down the hill", away from the local maximum and toward a global maximum.

The most famous algorithm which accepts non-improving solutions is *Simulated Annealing (SA)*. SA, which was first proposed in 1983 by Kirkpatrick, Gelatt and Vecchi [96], is based on the Metropolis-Hastings algorithm [80, 121]. In the Metropolis-Hastings algorithm the current candidate solution $s^k + \Delta s^k$ is selected over the previous candidate solution s^k if

$$\text{a randomly generated number in } [0,1] \leq \exp\left(\frac{f(s^k + \Delta s^k) - f(s^k)}{T}\right), \tag{2.19}$$

where $T > 0$ is a parameter called the *temperature*. If the current candidate solution is at least as fit as the previous solution, that is if $f(s^k + \Delta s^k) \geq f(s^k)$, then it will always be selected; whereas if it is less fit, then it is selected with a probability of $\exp\left(\frac{f(s^k+\Delta s^k)-f(s^k)}{T}\right)$.

SA differs from the Metropolis-Hastings algorithm in that the temperature is able to vary, typically as a function of the number of iterations and decreasing to zero as the number of iterations tends to infinity. The temperature decrease is analogous to the cooling of a material to into a minimum energy crystalline structure, a technique known as *annealing* (hence the name *Simulated Annealing*). A pseudocode description of SA on a continuous search space is given in Algorithm 2.8.

Algorithm 2.8: Simulated Annealing

Input : Fitness function f, search space $\mathcal{S}$, number of iterations N, temperature function $T(\cdot)$ and PDF $a(\cdot)$

Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
s* ← randomly generated point in S;
s^1 ← s*;
for k ← 1 to N do
    Δs^k ← randomly generated vector sampled from a(·);
    if randomly generated number in [0,1] ≤ exp((f(s^k+Δs^k)−f(s^k))/T(k)) then
        y^k ← 1;
        else y^k ← 0;
    end
    s^{k+1} ← s^k + y^k Δs^k;
    if f(s^{k+1}) > f(s*) then
        s* ← s^{k+1};
    end
end
```

The *cooling schedule*, which controls the rate at which the temperature decreases[9], comes in a

[9]Note that the temperature need not decrease and instead the system may be *reheated*.

variety of formats. A popular one is *geometric cooling* [22], for which

$$T(k+1) = \alpha T(k), \tag{2.20}$$

where $\alpha \in (0,1)$. Slower cooling can ensure the convergence of SA, such as that of Geman and Geman [63], who set

$$T(k) = \frac{c}{\log(1+k)}, \tag{2.21}$$

where $c \geq \delta$ and

$$\delta = \max_{s,t \in \mathcal{S}} \{f(s) - f(t)\} \tag{2.22}$$

denotes the maximum difference in fitness over the search space. The convergence of SA is one of its appealing properties. Based on the proof in [138], a simple proof of convergence for graphs with a minimum temperature[10] T_m, where $T(k) \geq T_m > 0$, is given below.

Lemma 2.5.1 ([138])**.** *Let d be the minimum degree of a graph $G = (V, E)$ and let D be the diameter of G. Futhermore, let $\hat{s}$ be a point of maximum, that is*

$$f(\hat{s}) \geq f(s) \quad \textit{for all} \ \ s \in \mathcal{S} \tag{2.23}$$

and let

$$\delta = \max_{i \in V, j \in N(i)} \{f(i) - f(j)\} \tag{2.24}$$

be the maximum fitness difference between a point and its neighbour. If the sequence of candidate solutions generated in the method described above is $\{s^k\}$, starting from any state s^1, then the expected number of steps before $\hat{s}$ is visited is at most $[d \exp(\delta/T_m)]^D + D$.

Proof. There exists a directed path from s^1 to $\hat{s}$ in G of length $q \leq D$. Let $e^1, e^2, \ldots, e^q$ be the sequence of edges in this path.

Clearly, the probability that $\hat{s}$ is visited, starting from s^1, is at least the probability that each one of the edges $e^1, e^2, \ldots, e^q$ is traversed in succession. The later probability is at least $\left[\frac{1}{d}\exp(-\delta/T_m)\right]^q \geq \left[\frac{1}{d}\exp(-\delta/T_m)\right]^D \equiv g$, assuming that each neighbour of a state is equally likely to be generated next.

Since $\{s^k\}$ forms a Markov chain, the probability that $\hat{s}$ is visited during the next q steps (for any q) does not depend on any of the states visited before. Let F be the expected number of iterations until $\hat{s}$ is visited. Then

$$\begin{aligned} F &= \sum_{i=1}^{\infty} [\text{Probability of path not begun during previous } i \text{ iterations}] + [\text{path length}] && (2.25)\\ &\leq \sum_{i=1}^{\infty} (1-g)^i + D && (2.26)\\ &= \frac{1}{g} + D \\ &= [d \exp(\delta/T)]^D + D, \end{aligned}$$

using in (2.25) the formula[11] $E[X] = \sum_{i=1}^{\infty} P(X \geq i)$. □

[10] It is reasonable to assume that there is a minimum temperature as it is always the case in practice.

[11] The addition of the term D in (2.26) was not included in the original proof — the author added the term for ease of understanding and is not necessary for the validity of the proof [145].

Theorem 2.5.2 ([138]). *Let $E = 2F$, where F is defined in (2.25). Then the SA algorithm converges in a time bounded above by kE, with probability at least $1 - 2^{-k}$, independent of the initial state.*

Proof. It is shown, by induction on k, that that the probability of $\hat{s}$ not being visited within kE steps is at most 2^{-k}.

Observe, as base case that, for any initial state s^1, the expected number of steps before $\hat{s}$ is visited is at most $E/2$ (using Lemma 2.5.1). An application of Markov's inequality[12] implies that the probability of $\hat{s}$ not being visited starting from s^1 within E steps is at most $1/2$.

Assume the hypothesis holds for all $k < (r-1)$. Let $s^E, s^{2E}, \ldots, s^{(r-1)E}$ be the states of the Markov chain during time steps $E, 2E, \ldots, (r-1)E$ respectively. Futhermore, let A be the event that $\hat{s}$ is not visited during the first E steps, and B be the event that $\hat{s}$ is not visited during the next $(r-1)E$ steps.

The probability that $\hat{s}$ is not visited within rE steps is $P = P(B|A)P(A)$. Using the fact that $\{s^k\}$ is a Markov chain, the conditional probability $P(B|A)$ depends only on what state the Markov chain is in at time step E and the time duration $(r-1)E$. Hence

$$\begin{aligned} P &= \sum_{i \in V \setminus \{\hat{s}\}} P(B|s^E = i) P(s^E = i) && (2.27) \\ &\leq \sum_{i \in V \setminus \{\hat{s}\}} 2^{-(r-1)} P(s^E = i) && (2.28) \\ &= 2^{-(r-1)} P(A) && (2.29) \end{aligned}$$

using the induction hypothesis that assumes $P(B|s^E = i) \leq 2^{-(r-1)}$ for each $i \in V \setminus \{\hat{s}\}$. But since $P(A) \leq 1/2$,

$$P \leq \frac{1}{2} 2^{-(r-1)} = 2^{-r}, \tag{2.30}$$

completing the induction step. □

There also exist proofs of the convergence for SA on continuous search spaces [48, 71, 106]. The novel proof presented here is similar to that for Matyas's algorithm, presented in Theorem 2.4.2.

Like in Matyas's algorithm, the step vector in the SA algorithm is assumed to be a Gaussian random vector with zero mean value and unit correlation matrix. The PDF $a(\cdot)$ of the step vector satisfies a slightly weaker relation than that for Matyas[13]; namely that there exist positive real numbers c and θ such that $a(s) > \theta$ for all $s \in \mathcal{S}$ satisfying $||s|| < c$.

Theorem 2.5.3. *Assume that $\mathcal{S}$ is strictly convex, that there exists some positive number r such that*

$$||s|| < r \text{ for all } s \in \mathcal{S} \tag{2.31}$$

and that $\mathcal{S}$ has a covering of open balls $B_i \subset \mathbb{R}^n$ of radius r_s each of which is contained in $\mathcal{S}$, that is

$$\mathcal{S} = \cup_{i \in I} B_i, \qquad B_i \subset \mathcal{S}. \tag{2.32}$$

Let $\hat{s}$ be the point of maximum for f in $\mathcal{S}$, that is

$$f(\hat{s}) \geq f(s), \quad \text{for all } s \in \mathcal{S}. \tag{2.33}$$

[12] Markov's inequality states that if X is any nonnegative random variable and $a > 0$, then $P(X \geq a) \leq E(X)/a$.
[13] The difference is that there exists a c, not for all c.

Furthermore,

$$R_\epsilon = \{s \in \mathcal{S} : |f(\hat{s}) - f(s)| < \epsilon\} \tag{2.34}$$

and let

$$h = \frac{1}{3}\min\{c, \epsilon, r_s\} \qquad \text{and} \tag{2.35}$$

$$D = \lceil 2r/h \rceil, \tag{2.36}$$

where $\lceil \cdot \rceil$ *is the ceiling function. Finally, let*

$$\mu_h = \int_{B(\cdot,h)} 1 dy \tag{2.37}$$

be the measure of a ball of radius h *in* $\mathbb{R}^n$ *(which is independent of its centre due to translational invariance) and let* δ *be the maximum fitness difference between two points, that is*

$$\delta = \max_{s,t \in \mathcal{S}} \{f(s) - f(t)\}. \tag{2.38}$$

Then the probability that R_ϵ *will be sampled starting from any point* $s^1 \in \mathcal{S}$ *is more than* $[\theta \mu_h \exp(-\delta/T_m)]^D$.

Proof. Let $q = \lceil ||\hat{s} - s_1||/h \rceil$, for which $0 < q \leq D$ by (2.36). Consider the sequence $z^2, z^3, \ldots, z^{q+1}$, defined by

$$\begin{aligned} z^k &= s^1 + (k-1)h\frac{\hat{s} - s^1}{||\hat{s} - s^1||}, \qquad k = 2, \ldots, q, \\ z^{q+1} &= \hat{s}. \end{aligned}$$

for which

$$||z^{k+1} - z^k|| \leq h, \qquad k = 2, \ldots, q. \tag{2.39}$$

Let $B^k \equiv B(z^k, h)$ represent the ball of radius h centered at z^k. Then B^k is contained in $\mathcal{S}$ for $k = 2, \ldots, q+1$, since $\mathcal{S}$ is strictly convex and

$$B^k \subset \left\{ s \in \mathcal{S} : \min_{s_L \in L(\hat{s}, s^1)} \{||s_L - s||\} < h \right\} \subset \mathcal{S} \tag{2.40}$$

by (2.35), where $L(\hat{s}, s^1)$ is the straight line between $\hat{s}$ and s^1.

The distance between any point in B^k and B^{k+1} is less than c, since for $x^k \in B^k$ and $x^{k+1} \in B^{k+1}$

$$||x^k - x^{k+1}|| \leq ||x^k - z^k|| + ||z^k - z^{k+1}|| + ||z^{k+1} - x^{k+1}|| \leq h + h + h \leq c \tag{2.41}$$

due to (2.35) and (2.39). Likewise, the distance between any point in B^2 and s^1 is less than c.

Consider the sequence $s^1, s^2, s^3, \ldots, s^q, s^{q+1}$ generated by the SA algorithm starting from s^1. The probability that $s^2 \in B^2$ is

$$P_{B^2}\{s^2\} = P_{B^2}\{s^1 + \Delta s \in B^2\} = \int_{B^2} a(y - s^1) dy \tag{2.42}$$

and the probability that $s^{k+1} \in B^{k+1}$ if $s^k \in B^k$ for $k = 2, \ldots, q$ is

$$P_{B^{k+1}}\{s^{k+1}\} = P_{B^{k+1}}\{s^k + \Delta s^k \in B^{k+1} : s^k \in B^k\} = \int_{B^{k+1}} a(y - s^k) dy, \tag{2.43}$$

where $a(\cdot)$ indicates the PDF of normal random vectors Δs.

From the properties of the PDF, (2.35) and (2.41) it may be concluded that

$$\inf_{y \in B^2} a(y - s^1) > \theta, \tag{2.44}$$

$$\inf_{y \in B^{k+1}, s^k \in B^k} a(y - s^k) > \theta \tag{2.45}$$

and then from (2.37), (2.42), (2.43), (2.44) and (2.45) that $P_{B^k}\{s^k\} > \theta\mu_h$ for all $k = 2, \ldots, q$. Therefore, the probability that the sequence $s^2, s^3, \ldots, s^q, s^{q+1}$ is generated in $B^2, B^3, \ldots, B^{q+1}$, respectively, is the product of the probability of each s^k being generated and accepted, that is

$$\begin{aligned} P &\geq \prod_{k=2}^{q+1} P_{B^k}\{s^k\} \exp(-\delta/T_m) \\ &> \prod_{k=2}^{q+1} \theta\mu_h \exp(-\delta/T_m) \\ &= [\theta\mu_h \exp(-\delta/T_m)]^q \\ &\geq [\theta\mu_h \exp(-\delta/T_m)]^D. \end{aligned}$$

Since $B^{q+1} \subset R_\epsilon$, by (2.34) and (2.35), it follows that the probability that R_ϵ will be sampled starting from s^1 is $> [\theta\mu_h \exp(-\delta/T_m)]^D$. □

Following the same argument as Lemma 2.5.1, it can be shown that the expected number of steps before $\hat{s}$ is visited is $F < [\theta\mu_h \exp(-\delta/T_m)]^{-D} + D$. Theorem 2.5.2 also follows, showing that SA converges no matter what the initial state is.

In summary, the SA algorithm is a stochastic search with acceptance of non-improving solutions. It is both fast and has guaranteed convergence, making it a powerful method. As a result of these qualities, as well as its simplicity, SA is used in many applications [22, 79, 81] and is probably the most prevalent trajectory method[14].

The class of trajectory methods represents only one of two types of metaheuristics, the other being *population* methods. The classical population method essentially employs multiple trajectory methods in parallel which exchange information every so many generations. In this way population methods are similar to multi-start trajectory methods, with the trajectories being run simultaneously instead of sequentially. Hence parallelism, populations and multi-start trajectory algorithms are all connected.

2.6 Parallelism and Populations

Luque *et al.* [111] describe how there are three major models for parallel trajectory-based metaheuristics, the *parallel exploration* of the neighbourhood, the *parallel evaluation* of each candidate solution, and the *multi-start* (or *island*) model. The first two models speed up the execution without changing the semantics of an algorithm, and are known as *master-worker*

[14] The prevalence of trajectory methods was roughly measured by searching for each method in Google Scholar (29/01/2013). The methods were found in the following numbers of articles: Guided Local Search, 2 150; Iterated Local Search, 3 370; GRASP, 4 570; Variable Neighbourhood Search, 5 680; Tabu Search, 67 100; Simulated Annealing, 531 000. Each method was searched for by finding articles with the exact phrase of its name, except for GRASP which was searched by finding articles with all of the words "Greedy Randomised Adaptive Search Procedures GRASP". Note, for the sake of comparisons, that Genetic Algorithms were found in 664 000 articles.

models [11]. The master (central processor) manages the selection and the replacement steps, while sending sub-populations to the workers (connected processors) to execute generating and evaluation tasks (in parallel). This is of benefit when the cost of generating or evaluating new candidate solutions is high.

The parallel multi-start model is interesting from an algorithmic point of view, since it exhibits different behaviour compared to its serial counterpart, random restart (see Section 2.3). It entails launching in parallel a *population* of several *independent* or *cooperative* subalgorithms[15]. Usually in a cooperative model each subalgorithm receives a candidate solution and engages in a search from that candidate solution. After a number of iterations there is an exchange of information (candidate solutions) via a central selection scheme and new solutions are received by each subalgorithm, after which the searches begin anew. The unavoidable trade-off is that information is lost either when a candidate solution is not chosen by the selection scheme and no new information is incorporated, or when it is accepted and the previous historical information of the subalgorithm is lost [111].

It is debatable whether populations and the exchange of information is beneficial. This is because, for a set amount of computing power, the number of function evaluations (which is usually the limiting factor) must be divided amongst the multiple subalgorithms. Hence, the number of iterations (corresponding to the number of function evaluations) for each subalgorithm is a fraction of that of a single algorithm run with the same computing power. Jansen and Wegener [90] offer an analysis of the use of populations and prove that they are beneficial for some problems, although they do mention that they are not applicable to other problems.

The next level of cooperation is not only exchanging information by means of solution selection, but using solutions from multiple subalgorithms to generate new solutions, a process known as *recombination*. There even exist algorithms that use information from all of the current solutions to generate solutions during each iteration (*e.g.* the Estimation of Distribution Algorithm).

There is a wide variety of population algorithms, including *Evolutionary Algorithms (EAs)*, *Differential Evolution* [136, 168], *Scatter Search* [67, 168], *Estimation of Distribution Algorithm* [100, 168], *Ant Colony Optimisation* [49, 168] and *Particle Swarm Optimisation* [130, 168]. The focus in the rest of this thesis is EAs, which are a specific type of population algorithm that mimics the principles of natural evolution.

2.7 Evolutionary Algorithms and chapter summary

This chapter is a novel narrative describing the development of EAs. It follows the natural progression of ideas from pure random search all the way to stochastic, accepting of non-improving solutions and population-based algorithms, in other words, EAs. The narrative progressed in the following order:

1. Pure random search is the simplest metaheuristic and guarantees convergence.
2. Simple hill climbing focuses the search around good solutions which enables incremental improvements, although the algorithm may experience premature convergence.

[15] In the paper by Baños *et al.* [11] the following distinction between multi-start and island models is made: "The multi-start paradigm consists of executing in parallel several local searches, without any information exchange ... The island-based models divide the entire population into several sub-populations distributed among different processors. Each processor is responsible for the evolution of one sub-population, and occasionally individuals migrate among islands." Island-based models can therefore be thought of as cooperative multi-start models.

3. Premature convergence may be countered by the following escape strategies: restarting local search, stochastic search and accepting non-improving solutions.
4. Theorem 2.4.1 for Rastrigin's fixed step size random search demonstrates that stochastic search is faster than pure random search.
5. Optimum and adaptive step size (including the 1/5-th success rule) was discussed.
6. Theorem 2.4.2 for the Matyas method shows that a stochastic search may guarantee convergence.
7. Lemma 2.5.1 and Theorems 2.5.2 and 2.5.3 demonstrate that accepting non-improving solutions may also guarantee convergence.
8. Populations enable the exchange of information between multi-start algorithms, potentially increasing computation speed.

This culminates in EAs, which combine the best features from the above list. They are hill climbing and consequentially are able to improve incrementally. On top of this, they employ all of the escape strategies to prevent premature convergence and may guarantee convergence[16].

It is highly unlikely that EAs were developed by researchers who followed the above-outlined investigation of metaheuristics, even though these concepts were surely present when they were first proposed around during the 1960s. Instead they were developed by mimicking the principles of natural evolution, as described in Chapter 1. Evolution (Universal Darwinism) has three components: *reproduction*, *evaluating fitness* and *selection*, which are incorporated into EAs. The escape strategies are inherent in EAs through stochastic reproduction, selection of non-improving solutions and populations. The pseudocode description of EAs can be seen in Algorithm 2.9 (evaluating fitness and selection are combined into the single operator $\texttt{select}(\cdot, f)$).

Algorithm 2.9: Evolutionary Algorithm

Input : Fitness function f, search space $\mathcal{S}$, number of iterations N, parent population size μ, offspring population size λ, reproduction operator $\texttt{reproduce}(\cdot)$, and selection operator $\texttt{select}(\cdot, f)$

Output: Point $s^* \in \mathcal{S}$ of approximate global maximum of f

```
P = {s_1, s_2, ..., s_μ} ← randomly generated set of points in S;
s* ← s_1;
for k ← 2 to N do
    P' = {s'_1, s'_2, ..., s'_λ} ← reproduce(P);
    P = {s_1, s_2, ..., s_μ} ← select(P, P', f);
    for j ← 1 to μ do
        if f(s_j) > f(s*) then
            s* ← s_j;
        end
    end
end
```

Traditionally the reproduction operator is actually two operators, recombination followed by *mutation*, although recombination is not required for an EA. The mutation and selection operators are examined in Chapter 3, while the recombination operators for GAs, EP and ES are investigated in later chapters.

[16] Examples of proofs of convergence may be found in references [146, 147].

CHAPTER 3

Principal Operators of Evolutionary Algorithms

Contents

In the preceding chapter it was established that EAs are stochastic, accepting of non-improving solutions and population-based algorithms. These characteristics are incorporated into the two principle operators of EAs: *mutation* and *selection*. Based on their biological counterparts, mutation and selection are the only two operators necessary, and arguably sufficient, for a successful EA [58, 133, 167].

3.1 Mutation

As discussed in the previous chapter, mutation is one of two reproduction methods used in EAs to generate new candidate solutions (the other being recombination). Mutation works in a similar manner to trajectory algorithms, by adding a step vector to each candidate solution and thereby *mutating* it. Although mutation has traditionally been seen as a "background" operator, it is arguably more important than recombination, and there exist highly successful algorithms that only use mutation and selection (known as *naive evolution*) [110, 133, 157, 167].

In accordance with the principles of Universal Darwinism (see Chapter 1), mutation should be used to generate *similar*, but not the exact same, individuals as previous candidate solutions. The similarity is essential for *inheritance*, without which Universal Darwinism fails and evolution does not progress.

Following from the example of Simulated Annealing (Section 2.5), mutation is a *stochastic* operator with a step vector PDF. This stochasticity ensures that the algorithm can produce a *variety* of candidate solutions, as is also required for Universal Darwinism.

The form of the PDF depends on the representation of the candidate solutions and metric of the search space. If the candidate solutions are vertices of a graph, then the PDF would be based on the distance between vertices, whereas if they are points in $\mathbb{R}^n$ then it may be determined by the Euclidean distance.

Different mutation operators can be constructed using different mechanisms of ensuring inheritance and creating variation. These are encoded in the PDFs used to generate step vectors, the most common of which are discussed in the following subsections.

3.1.1 Gaussian and Cauchy distributions

The *Gaussian distribution* (also known as the *normal distribution*) is a commonly used PDF in statistics. It is defined by

$$Gaussian(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right],$$

where μ is the *mean* (or *expectation*) and σ is the standard deviation (with σ^2 being the *variance*). Typically the mean is set to zero[1] and the standard deviation (which is proportional to the step size) is dictated by the specific problem at hand and may be adapted during the running of the algorithm (see Section 2.4.3). This distribution is used in EP and ESs [6] as well as in the Matyas method (see Section 2.4.4) and SA (see Section 2.5).

An example of a heavier tailed distribution is the *Cauchy* distribution,

$$Cauchy(x;\mu,\gamma) = \frac{1}{\pi}\left[\frac{\gamma}{(x-\mu)^2+\gamma^2}\right],$$

where γ is the *scale parameter* (similar to the standard deviation). The heavier tail is useful when the algorithm needs to escape a local maximum, as it has a higher probability of generating new candidate solutions far from the current candidate solution [75, 158].

Examples of both distributions are plotted in Figure 3.1. The fact that the PDFs are symmetric, unimodal and centered at the origin ensure both variance and inheritability. The Cauchy distribution clearly has a heavier tail, increasing the likelihood of generating of distant candidate solutions.

Spherical symmetry of distributions

Multidimensional problems may, or may not, require that the PDF is spherically symmetrical. Consider the PDF generated by applying distributions along two Cartesian coordinates, x and y, for which the Gaussian distribution becomes

$$Gaussian(x,y;\mu_x,\mu_y,\sigma_x,\sigma_y) = \frac{1}{\sigma_x\sigma_y 2\pi} \exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}\right]$$

and the Cauchy distribution is given by

$$Cauchy(x,y;\mu_x,\mu_y,\gamma_x,\gamma_y) = \frac{1}{\pi^2}\left[\frac{\gamma_x}{(x-\mu_x)^2+\gamma_x^2}\right]\left[\frac{\gamma_y}{(y-\mu_y)^2+\gamma_y^2}\right].$$

[1] An exception to this is the bias vectored algorithm of Matyas, discussed in [23].

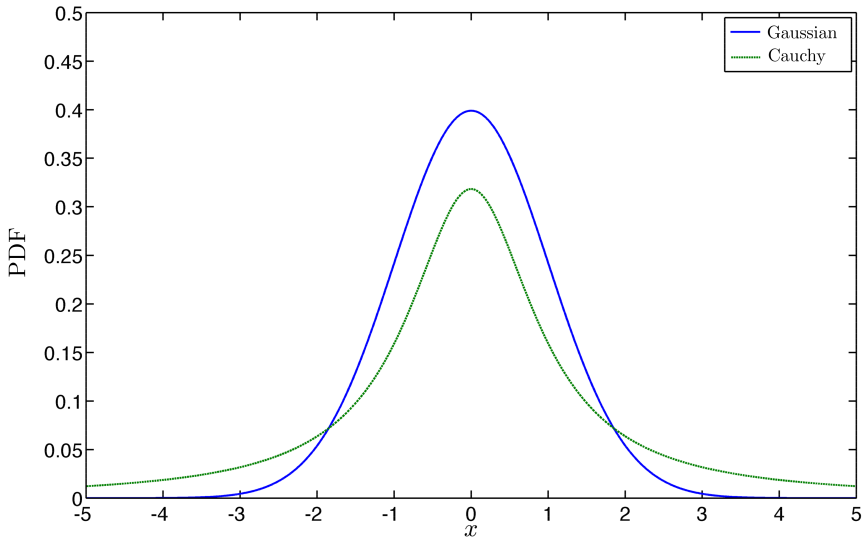


FIGURE 3.1: *Plot of the Gaussian and Cauchy distributions with zero mean, unit standard deviation and unit scale parameter.*

Assuming that the same distribution is applied to both Cartesian coordinates centered at the origin (*i.e.* $\mu_x = \mu_y = 0$, $\sigma_x = \sigma_y \equiv \sigma$, $\gamma_x = \gamma_y \equiv \gamma$) and converting into spherical coordinates ($x = r\cos\theta, y = r\sin\theta$) results in the following distributions,

$$\begin{aligned} Gaussian(r,\theta;\sigma) &= \frac{1}{\sigma^2 2\pi}\exp\left[-\frac{r^2}{2\sigma^2}\right] \quad \text{and} \\ Cauchy(r,\theta;\gamma) &= \frac{1}{\pi^2}\left[\frac{\gamma^2}{r^4\sin^2(2\theta)/4 + r^2\gamma^2 + \gamma^4}\right], \end{aligned}$$

from which it is evident that the Gaussian distribution is spherically symmetric whereas the Cauchy distribution is not. The lack of spherical symmetry may be countered by certain measures, such as using one distribution to generate the radius and another to generate the angle(s).

3.1.2 Binary code

For certain problems it may be convenient to represent candidate solutions in binary code [37, 116]. In the case of continuous optimisation problems for which binary code is used, a conversion between real numbers and binary code is required for evaluating the fitness of candidate solutions. A disadvantage of this approach is that binary code is limited accuracy due limited number of bits (compared to floating point) and that the binary distance between two numbers does not correlate exactly with their real distance.

The standard measure of the distance between two binary numbers $u = (u_1, u_2, \ldots, u_k)$ and $v = (v_1, v_2, \ldots, v_k)$ is the *Hamming distance*. The *Hamming vector* is defined by $h_i(u,v) = |u_i - v_i|$ and the Hamming distance is given by $h(u,v) = \sum_{i=1}^{k} h_i(u,v)$.

The relationship between the Euclidean and Hamming distance can be seen in Figure 3.2. It is clear from the line of best fit that there is a postive correlation, although this is not always respected. The aberations are known as *Hamming cliffs* and occur when the Euclidean distance is small between two binary numbers whose Hamming distance is large. For example, consider the numbers[2] $01111_2 = 15$ and $10000_2 = 16$, between which the Euclidean distance is 1, yet the Hamming distance is 5.

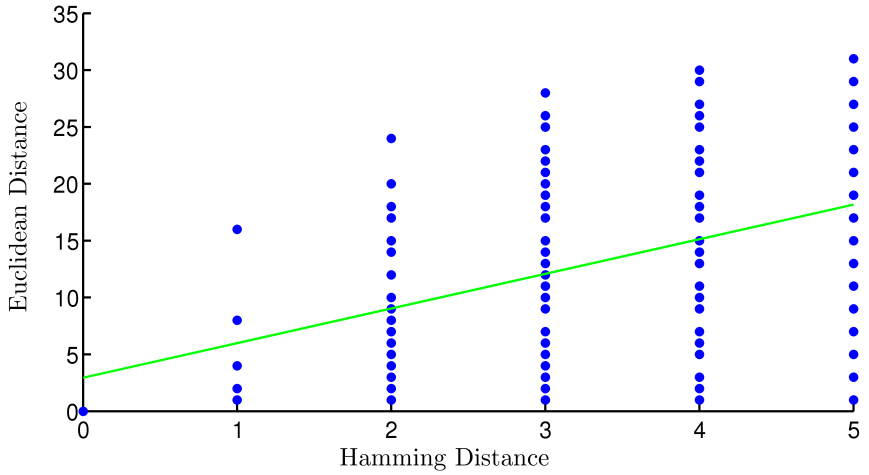


FIGURE 3.2: *Plot showing the correlation between the Euclidean and Hamming distance, computed for all combinations of binary numbers of length five. The dots represent the points at which the Euclidean and Hamming distances are equal and the line is that of best fit.*

Grey code

The problem of Hamming cliffs may be overcome by employing *Grey code* [99]. This code is designed to ensure that the Euclidean distance is one between two binary numbers whose Hamming distance is one. The construction of the code is done recursively by reflecting a list, then concatenating the original list with the reversed list, prefixing the entries in the original list with 0 and those in the reflected list with 1. To demonstrate how this is done, the 3-bit list may be generated from the 2-bit list as follows:

2-bit list	00,01,11,10
Reflect	10,11,01,00
Prefix 2-bit list with 0	000,001,011,010
Prefix reflected list with 1	110,111,101,100
Concatenate	000,001,011,010,110,111,101,100.

It is useful to visualise Grey code in the form of a pie chart. The 4-bit grey numbers are illustrated in Figure 3.3. Like for standard binary numbers, there is a correlation between the real value of a number in Grey code and its Hamming Distance, as may be seen in Figure 3.4.

[2]The '2' in the subscript indicates that the number is in binary form.

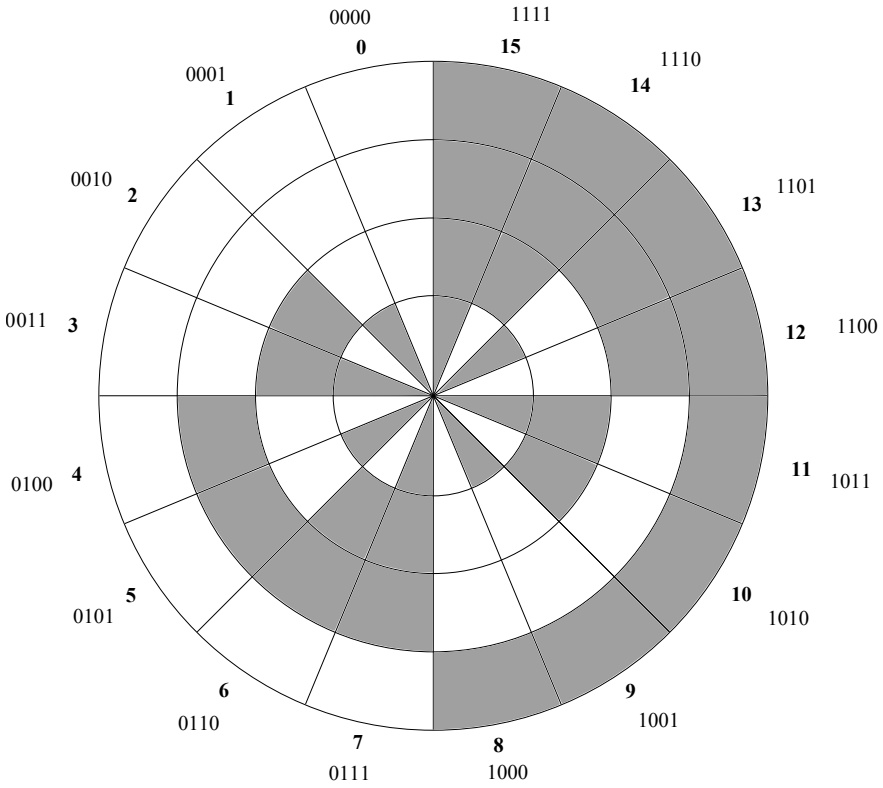


FIGURE 3.3: *Pie chart illustrating the 4-bit Grey numbers, with the grey shading indicating a 1 in the respective bit (with white representing a 0).*

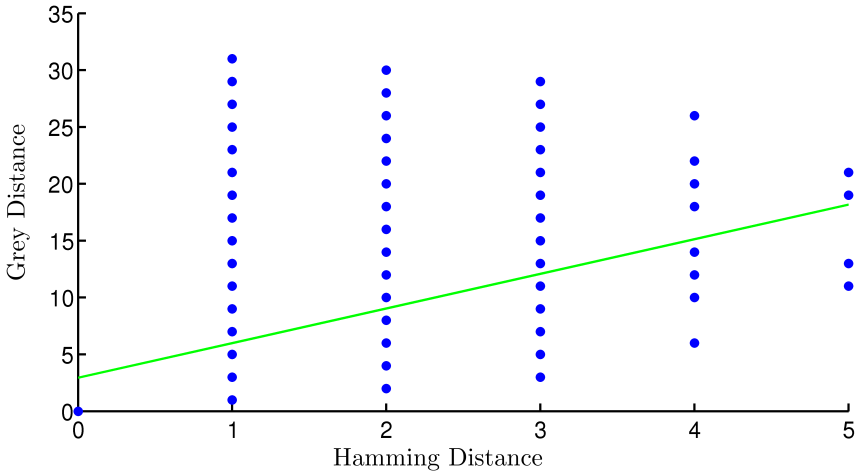


FIGURE 3.4: *Plot showing the correlation between the Grey and Hamming distance, computed for all combinations of binary numbers of length five. The dots represent the points at which the Grey and Hamming distances are equal and the line is that of best fit.*

Mutation distribution

Mutation takes the previous candidate solution, known as the *parent*, to produce the new candidate solution, known as the *child*[3]. Mutating binary numbers is achieved by *flipping* each bit (that is, changing it from 0 to 1 or from 1 to 0) with a certain probability, p_m. This is equivalent to adding a random binary mutation vector (with each bit having value 1 with a probability of p_m) modulo 2 to a parent to create a child. An example of this is:

$$\begin{array}{rcl} \text{Parent} & & 0011 \\ \text{Mutation Vector} & +_2 & \underline{1001} \\ \text{Child} & = & 1010. \end{array}$$

Although the generation of the binary mutation vector is independent of the parent, the step vector depends on the parent. To see this, consider the individuals $v = 11 = 1011_2$ and $w = 15 = 1111_2$, which are both mutated by adding the mutation vector $m = 0101_2$. In the equations below, the top line in each bracket shows binary addition, whereas the bottom line shows the addition of the step vector:

$$\begin{aligned} v + m &= \begin{cases} 1011_2 + 0101_2 & = 1110_2 \\ 11{+}\mathbf{3} & = 14 \end{cases}, \quad \text{while} \\ w + m &= \begin{cases} 1111_2 + 0101_2 & = 1010_2 \\ 15{-}\mathbf{5} & = 10 \end{cases}. \end{aligned}$$

Clearly, the same mutation vector m may result in completely different step vectors, $+3$ and -5. This is because the mutation vector only specifies which bits are to be flipped, not the direction of the flip (0 to 1 or 1 to 0), which is specified by the parent.

Heavy-tailed mutation distribution

The PDF of a binary mutation is discrete and exhibits large discontinuities due to Hamming Cliffs. It is useful for analytic purposes to approximate it by a continuous distribution, which may be achieved by the following novel procedure.

It is observed from the binary mutation distribution $\mathcal{M}$ of the zero vector (displayed in Figure 3.5) that

$$\begin{aligned} \mathcal{M}(2^{n+1} + x) &= \mathcal{M}(2^n + x) \quad \text{and} \\ \mathcal{M}(2^{n+1} + 2^n + x) &= p_m \mathcal{M}(2^n + x), \end{aligned}$$

where $x \in (0, 2^n)$ and n is a natural number. Integrating from 0 to 2^n on both sides and adding yields

$$\int_0^{2^n} \mathcal{M}(2^{n+1} + x)dx + \int_{2^n}^{2^{n+1}} \mathcal{M}(2^{n+1} + x)dx = \int_0^{2^n} \mathcal{M}(2^n + x)dx + p_m \int_0^{2^n} \mathcal{M}(2^n + x)dx,$$

which may be simplified to

$$\int_0^{2^{n+1}} \mathcal{M}(2^{n+1} + x)dx = (1 + p_m) \int_0^{2^n} \mathcal{M}(2^n + x)dx.$$

[3]This terminology is common and is taken from the biological counterpart. The term "offspring" is also sometimes used for "child" and candidate solutions are also known as "individuals".

This may be notationally condensed to

$$\overline{\mathcal{M}(2^{n+1},2^{n+2})}2^{n+1} = (1+p_m)\overline{\mathcal{M}(2^n,2^{n+1})}2^n,$$

or alternatively

$$\overline{\mathcal{M}(2^{n+1},2^{n+2})} = \frac{1+p_m}{2}\overline{\mathcal{M}(2^n,2^{n+1})},$$

where $\overline{\mathcal{M}(a,b)} = \int_a^b \mathcal{M}(x)dx/(b-a)$ is the average value over the interval.

This suggests a function of the form $\mathcal{M}(2x) = \left(\frac{1+p_m}{2}\right)\mathcal{M}(x)$. Assuming that $\mathcal{M}(x) = ax^{\alpha-1}$, it follows that $\alpha = \log_2(1+p_m)$ and the PDF is given by

$$\mathcal{M}(x;p_m) = ax^{\log_2(1+p_m)-1},$$

where $a = \log_2(1+p_m)/\left(2^n-1\right)^{\log_2(1+p_m)}$ normalises the PDF such that the total probability is one. The PDF may be approximated as $\mathcal{M}(x;p_m) \approx ax^{p_m/\ln(2)-1}$ for $p_m \ll 1$, or even further approximated using infinite series,

$$\begin{aligned}
\mathcal{M}(x;p_m) &= \frac{a}{x^{1-\log_2(1+p_m)}} \\
&= \frac{a}{x+\sum_{i=1}^{\infty} x\left[-\log_2(1+p_m)\ln x\right]^i/i!} \\
&= \frac{a}{x}\cdot\sum_{j=0}^{\infty}\left((-1)^j\sum_{i=1}^{\infty}\left[-\log_2(1+p_m)\ln x\right]^i/i!\right)^j \\
&\approx \frac{a}{x},
\end{aligned}$$

which is valid for $\max(|x|,\left|x^{1/x}\right|) \ll \exp(1/\log_2(1+p_m))$.

It is difficult to assess the agreement of the continuous PDF with the binary PDF. A simple comparison may be made by plotting the cumulated PDF for both cases, as is done in Figures 3.5 to 3.8, which show that the continuous PDF and binary PDF are very similar[4]. Due to the discrete nature of binary code the distributions diverge at certain points, depending on the candidate solution. However, if averaged over all candidate solutions[5] (as done in Figures 3.7 and 3.8), the distributions are smoothed and it appears that the binary distribution is slightly lighter-tailed than the continuous distribution. As the p_m value tends to zero the probability of having two mutations opposed to one becomes zero and the distribution becomes dominated by one bit mutations. In this case the distributions intersect exactly at the points of one bit mutation (*i.e.* for powers of two), as displayed in Figure 3.8. Since the standard p_m values are very small [6], it may be assumed that the distributions share the same shape, specifically the same tail-heaviness.

Because the continuous PDF is not exponentially bounded it is heavy-tailed, hence, the binary mutation PDF is heavy-tailed. As may be expected, in multiple dimensions the binary PDF is not spherically symmetric (see Section 3.1.1), with $\mathcal{M}(r,\theta;p_m) = a(r^2\sin(2\theta)/2)^{\log_2(1+p_m)-1}$.

Note that the continuous PDF is also comparable to Grey code, as illustrated in Figures 3.9 to 3.12. Although the distributions are similar, it does not hold that they intersect exactly as p_m value tends to zero. It may be concluded that the Grey PDF has a slightly lighter tail than the continuous PDF.

[4] A high p_m value was used for illustrative purposes, but the agreement is strong for all values of p_m and number of bits.

[5] The averaged cumulative probability figures display values up to 0.5, not 1, since only on one side of the distribution is shown (the distribution is symmetric).

Binary PDF and its continuous approximation figures

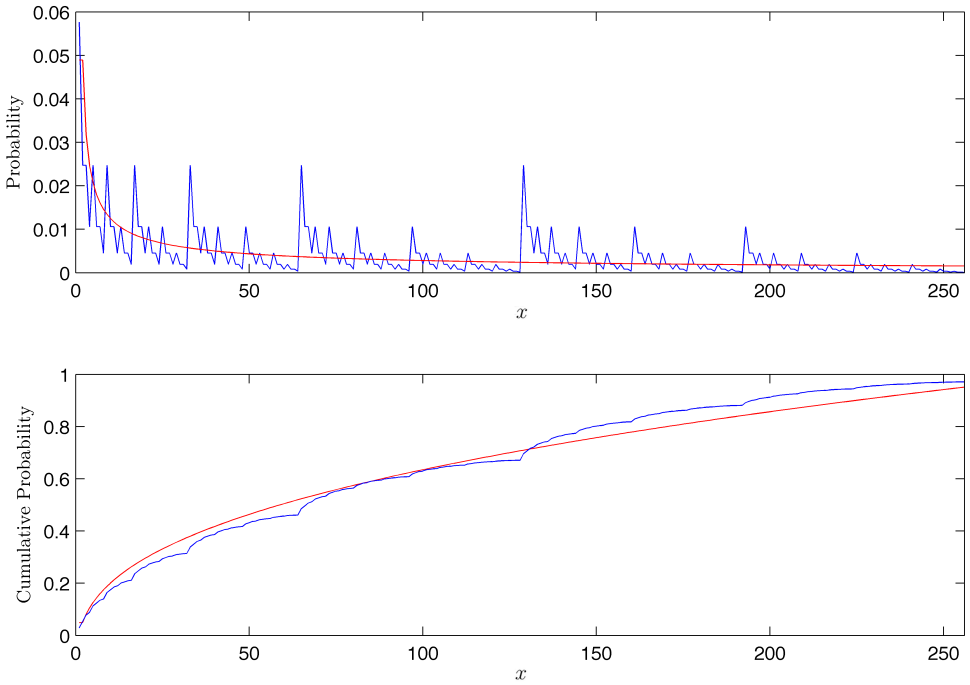


FIGURE 3.5: *The probability distribution for the candidate solution* 00000000_2 *with* $p_m = 0.3$.

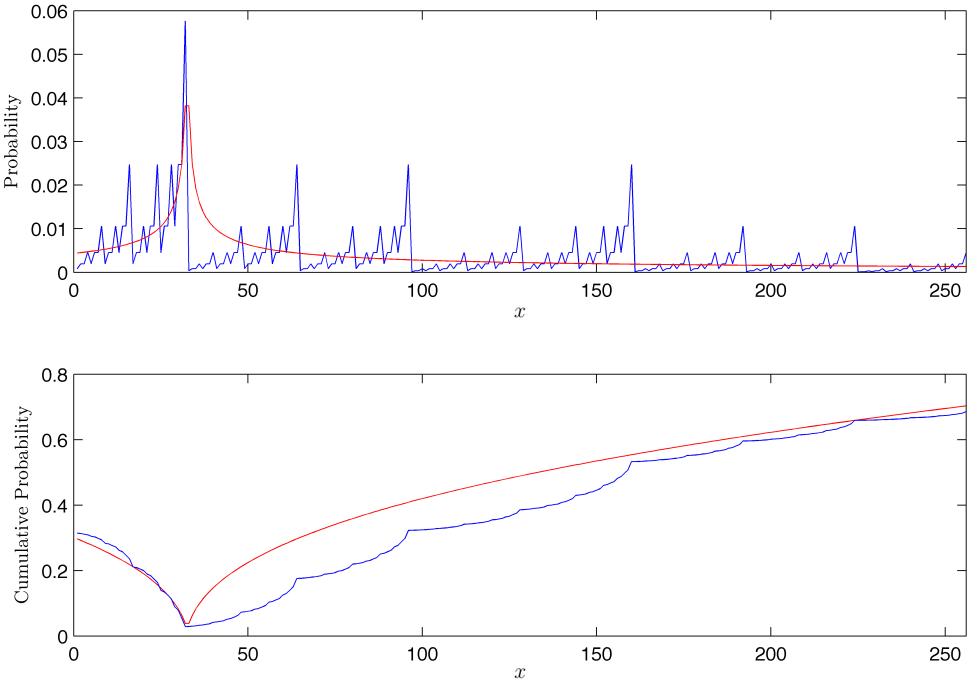


FIGURE 3.6: *The probability distribution for the candidate solution* 11111000_2 *with* $p_m = 0.3$.

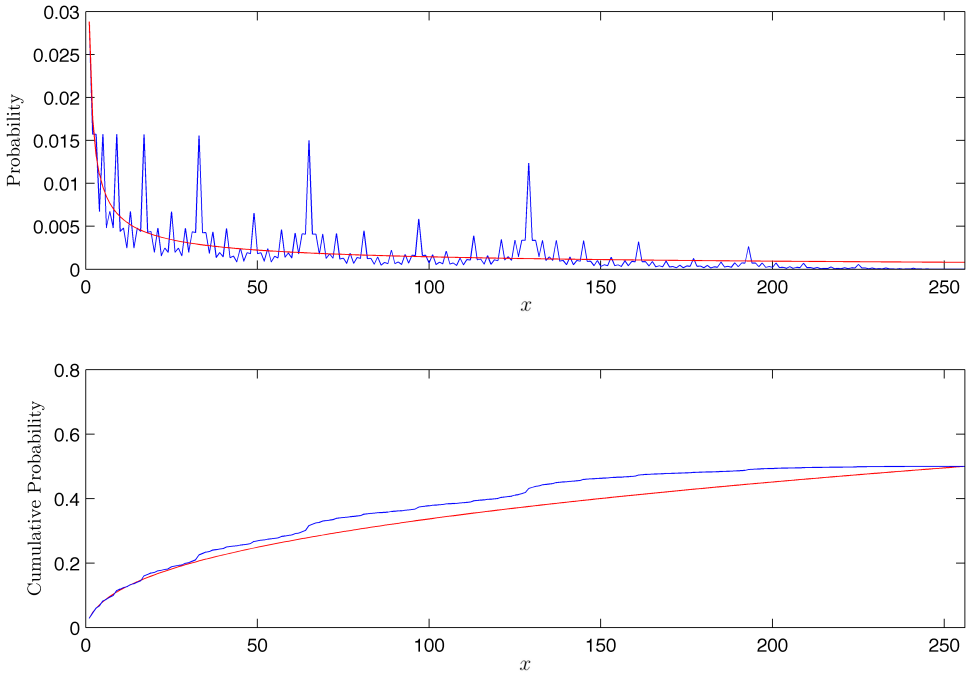


FIGURE 3.7: *The probability distribution averaged over all candidate solutions with* $p_m = 0.3$.

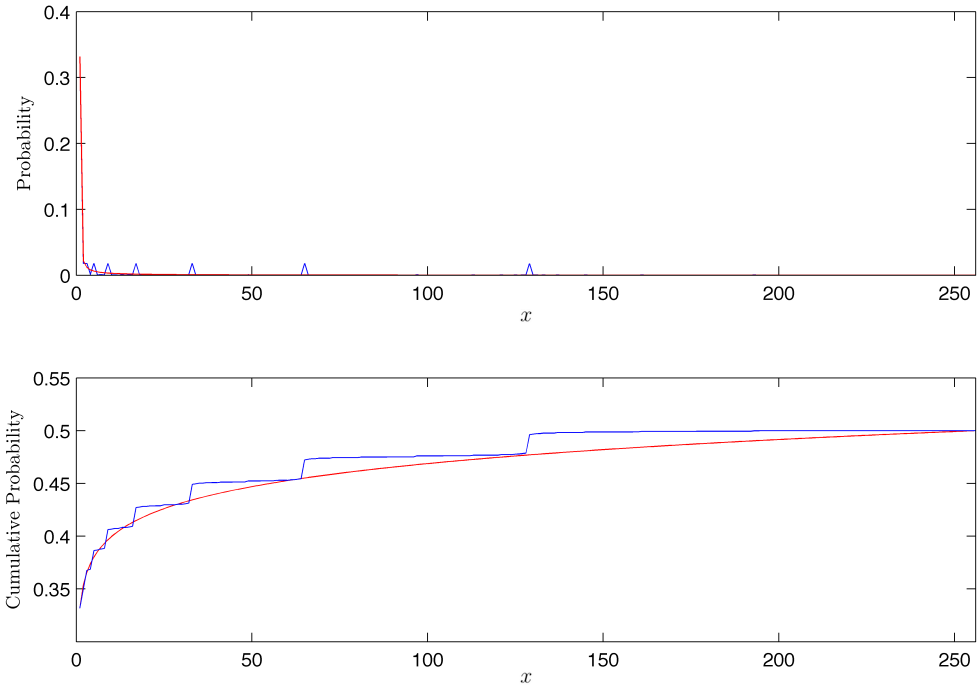


FIGURE 3.8: *The probability distribution averaged over all candidate solutions with* $p_m = 0.05$.

Grey PDF and its continuous approximation figures

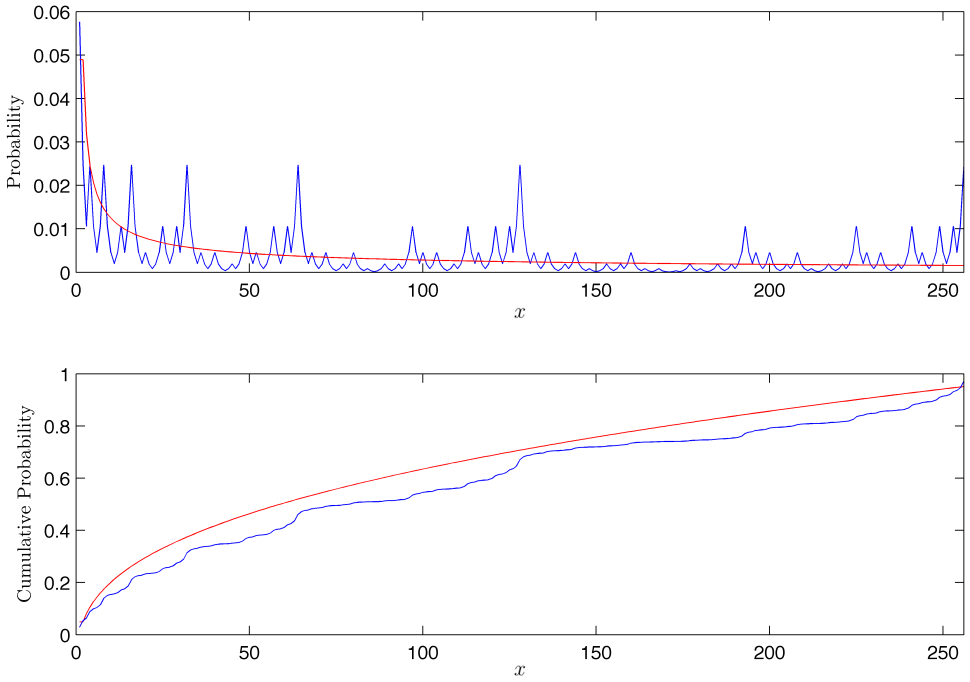


FIGURE 3.9: *The probability distribution for the candidate solution* 00000000_2 *with* $p_m = 0.3$.

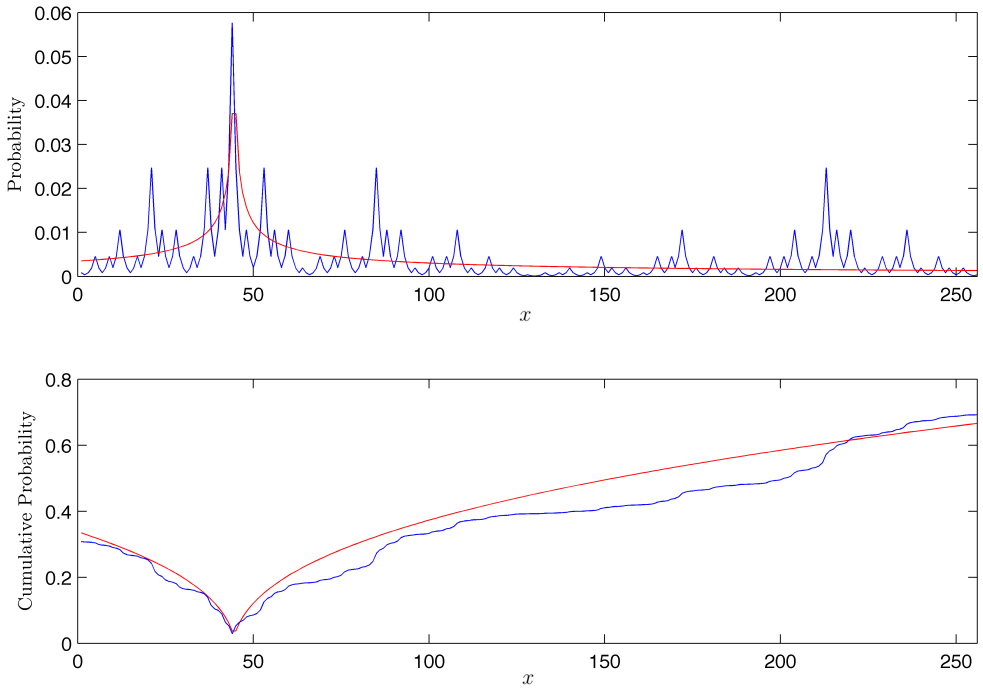


FIGURE 3.10: *The probability distribution for the candidate solution* 00111110_2 *with* $p_m = 0.3$.

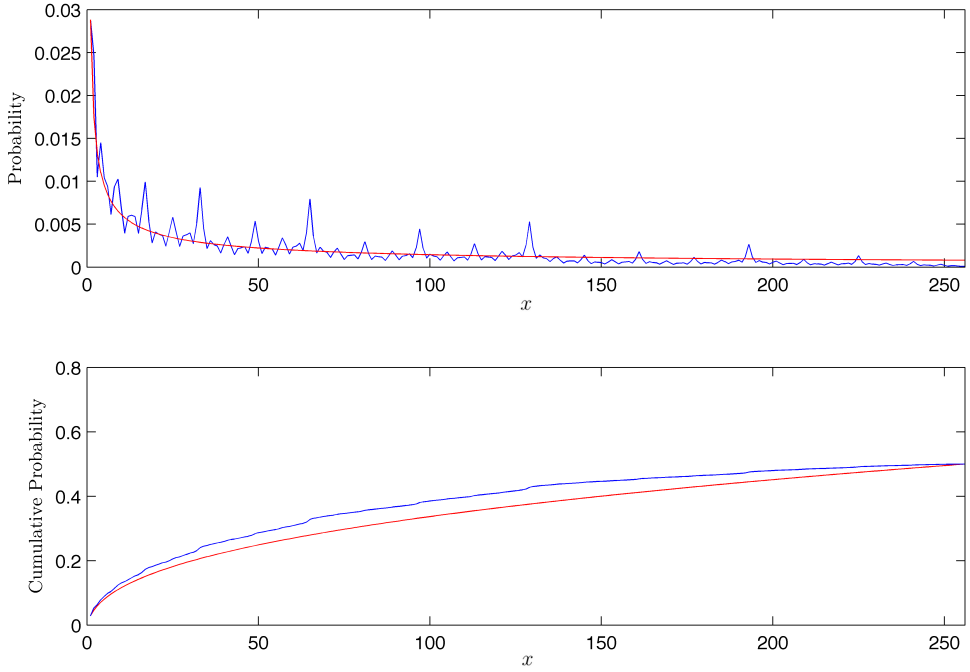


FIGURE 3.11: *The probability distribution averaged over all candidate solutions with* $p_m = 0.3$.

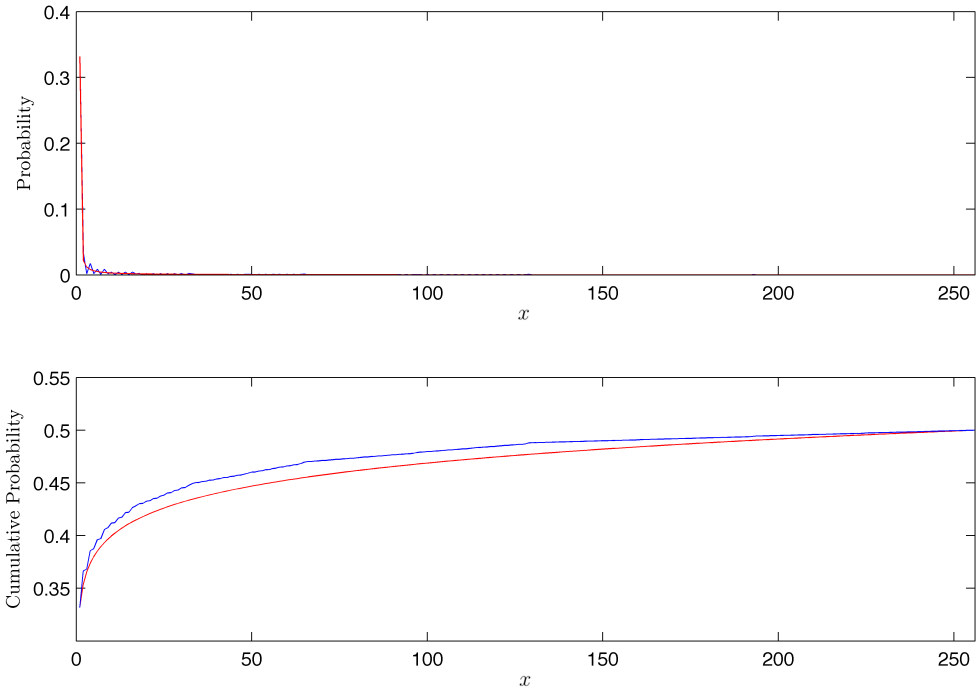


FIGURE 3.12: *The probability distribution averaged over all candidate solutions with* $p_m = 0.05$.

3.2 Selection

The only requirement for an EA selection scheme is that candidates of higher fitness are more likely to be selected during the next generation. There are a number of questions raised in developing a selection scheme, such as: which candidate solutions should be considered for selection, should there be deterministic elements to selection, how strong should *selection pressure*[6] be and, most generally, how should fitness values be converted into selection probabilities? A wide variety of schemes have been suggested, some of which are historically associated with a particular algorithm. Similarly to mutation operators (and all EA operators for that matter), it is impossible to show that one scheme is generally better than another, although some have proven to be more popular (a technical comparison of methods can be found in [7, 45, 70, 73] according to [122]). The most common schemes used in EAs are revealed in this section.

3.2.1 Preselection, niching and crowding

Preselection essentially implements a similar selection scheme to SA (see Section 2.5) run in parallel and thereby avoids the problem of having to select from a large set of multiple individuals. If a child has higher fitness than its worse parent, then it replaces that parent (if not, then the child may be discarded or selected according to a certain probabilistic scheme) [28, 112]. Since recombination may use multiple parents to generate a child, preselection differs from selection in parallel trajectory algorithms.

The appeal of preselection is that it maintains population diversity. In order for the diversity of a population to be maintained, newly generated individuals should replace individuals in the current population that are most similar to them [184]. As parents are usually similar to their children, it follows that they are good candidates for replacement. Hence, dissimilar individuals do not compete and instead occupy their own *niche*, as is the case in biology (a selection scheme that enables this behaviour may be described as *niching*).

This notion was extended by De Jong [93], who proposed *crowding*. With crowding a newly generated individual is compared to a sample individuals randomly drawn from the population. The new individual replaces the individual in the sample that is most similar to it.

3.2.2 Steady state and generational replacement schemes

The above selection schemes are known as *steady state* schemes, since only a small number of offspring (one or two) replace an equal number of parents to form the population of the next generation [51]. This is in direct contrast to *generational replacement*, in which the next generation is entirely composed of offspring from the previous generation. An intermediate between these two schemes is the selection of the next generation from the union of offspring and parents. Such selection may include *elitism*, which ensures that certain parents of high fitness are selected.

The following notation is standard in describing the type of selection scheme with a parent population size of μ and an offspring population size of λ: algorithms that select only from the offspring population are denoted as (μ, λ)-algorithms, whereas algorithms that select from the union of the parent and offspring population are denoted as $(\mu + \lambda)$-algorithms.

[6] The selection *pressure*, *strength* or *intensity* refers to the expected average fitness of the population after selection [21].

3.2.3 Truncation, deterministic and stochastic schemes

A *truncation* scheme simply selects the μ fittest individuals from a population. Its determinacy gives it the advantage of faster computation time and conceptual simplicity. However (as was discussed in Section 2.5), accepting non-improving solutions is a useful method for escaping local maxima. Hence, most selection schemes are stochastic, with fitter individuals only being more likely to be selected.

3.2.4 Fitness proportional selection

The name "fitness proportional selection" is fairly self-explanatory — the expected number of selections (or probability of being selected) is proportional to the fitness of an individual. In John Holland's original selection scheme [51, 122], the expected number of selections E_i of the candidate solution s_i is

$$E_i = \frac{\mu}{\sum_{j=1}^{\Omega} f(s_j)} f(s_i),$$

where μ individuals are selected from a population of size Ω. This scheme may be implemented either by *Roulette Wheel Selection (RWS)* or *Stochastic Universal Sampling (SUS)*.

The metaphor for both of these schemes is based on a roulette wheel, as represented in Figure 3.13. A roulette wheel is divided into Ω segments, representing the Ω individuals in the population, with the size of each segment being proportional to the fitness of the respective individual.

In SUS, the wheel has μ evenly spaced *selection pointers* and the wheel is only spun once, with each pointer picking out an individual for the next generation. SUS is given in pseudocode form[7] as Algorithm 3.1.

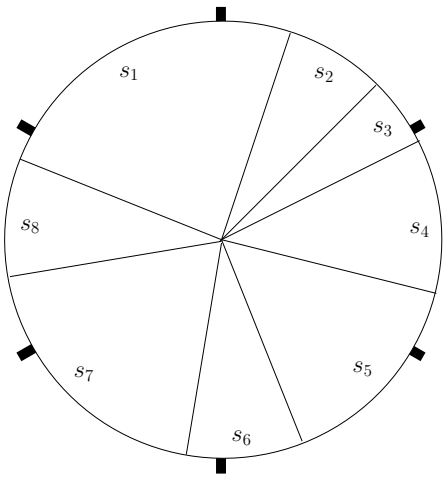


FIGURE 3.13: *A roulette wheel representing SUS. There are six selection pointers evenly spaced around the wheel and eight possible individuals $s_1, \ldots, s_8$ to select from. As the wheel stands, the following individuals are selected: $s_1, s_1, s_3, s_5, s_6, s_7$.*

[7] In the code the notion of a cumulative sum is used. The i^{th} element of the cumulative sum of $f(s_1), \ldots, f(s_\Omega)$ is equal to $\sum_{j=1}^{i} f(s_j)$.

Algorithm 3.1: Stochastic Universal Sampling

Input : Fitness function f and population $s_1, \ldots, s_\Omega$
Output: Next generation

$(c_1, \ldots, c_\Omega) \leftarrow$ cumulative sum of the fitness values $f(s_1), \ldots, f(s_\Omega)$;
$p \leftarrow$randomly generated number in $(0, c_\Omega)$;
for $k \leftarrow 1$ **to** μ **do**
    Select s_i where i is the smallest integer such that $c_i > p + kc_\Omega/\mu \pmod{c_\Omega}$
end

In RWS the wheel has only one selection pointer. After the wheel has been spun, the pointer picks out one individual which is added to the next generation, and this is repeated μ times until the entire next generation is selected. RWS is given in pseudocode form as Algorithm 3.2.

Algorithm 3.2: Roulette Wheel Selection

Input : Fitness function f and population $s_1, \ldots, s_\Omega$
Output: Next generation

$(c_1, \ldots, c_\Omega) \leftarrow$ cumulative sum of the fitness values $f(s_1), \ldots, f(s_\Omega)$;
for $k \leftarrow 1$ **to** μ **do**
    Select s_i where i is the smallest integer such that c_i exceeds a randomly generated number in $(0, c_\Omega)$
end

The advantage of SUS over RWS is that the individual with the highest fitness is guaranteed to be selected and that there is less variance (noise) [9] (according to [6, 51]). For example, it is possible for the individual with the lowest fitness to be repeatedly selected by RWS, but may be selected at most once by SUS.

Problems arise when the fitness function is not well scaled for selection. For example, if the fitness values are all large then the relative fitness difference is small and each individual has a similar probability of being selected. In this case subtracting a constant amount from each fitness value would increase the relative fitness difference, and thereby make selection more differentiating. There are many types of fitness scaling, including subtracting a constant or elevating the fitness to an adequate power k (with k proportional to the selection pressure) [51].

The problem of fitness scaling can be avoided using *linear ranking*. Each of the individuals are ranked from most fit (first) to least fit (last) and are accorded a ranked fitness $f_r = 1 - r/\Omega$, which may then be used (instead of the actual fitness values) for selection. To alter the selection pressure, the ranked fitness values may still be manipulated via fitness scaling.

3.2.5 Tournament selection

Tournament selection completely sidesteps the issue of fitness scaling and effectively implements a type of rank selection. It entails running several tournaments in which a random set of individuals are drawn from the population to compete, with only the winner being selected for the next generation. The two parameters are the tournament size T and probability of the individual in the tournament with the highest fitness being selected p. For the other individuals in the tournament, the one with the second highest fitness is selected with probability $p(1-p)$, the third highest with probability $p(1-p)^2$, ... , the second lowest with probability $p(1-p)^{T-2}$

and the lowest fitness with probability $(1-p)^{T-1}$. If $p = 1$ then the individual with the highest fitness is always selected, in which case the tournament is called *deterministic*; if not, it is called *stochastic* [51]. The selection pressure may be controlled by altering T and p, with greater values corresponding to greater pressure. Tournament selection is given in pseudocode form as Algorithm 3.3.

Algorithm 3.3: Tournament selection

Input : Fitness function f, population $s_1, \ldots, s_\Omega$, tournament size T and selection probability p

Output: Next generation

```
for j ← 1 to μ do
    (t_1, ..., t_T) ← random subset of the population ordered such that f(t_i) > f(t_{i-1});
    for k ← 1 to T do
        if no individual has been selected for this j value then
            Select individual t_k with probability p(1 − p)^{k−1}, or with probability 1 if k = T
        end
    end
end
```

3.3 Chapter summary and simple EAs

Mutation and selection are all that is required to form a simple EA. The mutation operator should be chosen such that it has the appropriate PDF; with reasonable variance, tail heaviness and spherical symmetry if necessary. Two types of continuous mutation operators are presented and analysed in this chapter, Gaussian and Cauchy distributions, as well as the binary mutation distribution. Next a satisfactory selection operator must be decided upon. It must have a suitable selection pressure, may be steady state or generational, and deterministic or stochastic. Incorporated into the selection operator is the parent population size, offspring population size and from what combination of these two populations the next generation is to be selected from. The options for selection operators include preselection, truncation selection, fitness proportional selection and tournament selection. Together with the representation of the problem and stopping criteria (such as number of iterations for which the algorithm is run), these constitute the factors that need to be taken into account when setting up a simple EA.

How are the appropriate operators to be chosen? As explained in the opening chapter, little progress has been made in theoretically calculating the optimum operator for a certain problem. Empirical results may suggest problem classes in which certain operators do well. However, the empirical results themselves are not sufficient to explain *why* an operator performs well. The following question is considered in the next chapter: How are explanations employed to analyse empirical results?

CHAPTER 4

The Probable Fitness Landscape

Contents

Empirical simulations of problem instances may be used to demonstrate the performance of an algorithm in a particular problem class [89]. In fact, recently a new empirical methodology has managed to show that certain metaheuristics have superior performance in the problem class of binary real-world problems [62]. Although empirical results indicate expected runtime performance, they do not explain why an algorithm performs as it does. As Cohen [38] puts it (according to [175]): "It is good to demonstrate performance, but it is even better to explain performance."

To complement empirical analysis, qualitative descriptions may be used to explain the performance of an algorithm. The most common terminology used in these explanations includes:

exploitation, *exploration*, *intensity* and *diversity*. These terms are used extensively in the literature, appearing in the vast majority of articles in the leading journals on metaheuristics. Hence, these terms are of great importance in the field of metaheuristics, and EAs in particular.

However, there are no universally agreed upon definitions of these terms [54, 128]. Specific definitions may be given in each context of use, but these must be shown to be meaningful and consistent with the rest of the literature in order to be generally effective. If the terminology is meaningless or inconsistently applied then it loses its communicative power and thereby undermines its use in research publications.

In this chapter, six of the prevalent views on exploitation and exploration in the literature are identified and it is argued that they are all meaningful and consistent. This is a direct consequence of them all being derivable from novel definitions of exploitation and exploration, also proposed in this chapter. In turn, these definitions are based on a hypothetical construct, the *Probable Fitness Landscape* (PFL), which is also presented. A limitation is that the definitions only apply when the PFL is applicable, that is, for continuous fitness functions. Since continuity is arguably common to all metaheuristics [62, p.2129], this limitation is not too restrictive.

The PFL may also be used as a basis for the *Ideal Probability Distribution* (IPD), which, as its name suggests, is a hypothetical ideal probability distribution for generating individuals. In combination with diversity considerations and computational speed, the IPD may be used to explain the performance of an algorithm.

The chapter is structured as follows: In Section 4.1 the current use of the terms *exploitation, exploration, intensity* and *diversity* in the literature is reviewed. The PFL is introduced in Section 4.2 and is used to formally define the notions of exploitation and exploration, from which the prevalent views on exploitation and exploration are deduced in Section 4.3. The IPD and diversity are investigated in Section 4.5 and, using these tools, the principle EA operators are revisited for analysis purposes in Section 4.6. A summary of the material of this chapter may be found in Section 4.7.

4.1 Literature review

In *Review of Metaheuristics and Generalized Evolutionary Walk Algorithm*, Yang [187, p.3] states that "the main components of any metaheuristic algorithms are: intensification and diversification, or exploitation and exploration." These terms are certainly extensively used in the literature. The *Journal of Heuristics*, *IEEE Transactions on Evolutionary Computation* and *Evolutionary Computation*, three of the leading journals in the field of metaheuristics, respectively referred to these terms (and their derivatives) in 81%, 91% and 71% of papers in 2011 and 2012 (see the appendix at the end of the thesis for details). The frequency of use of the terms varies. In all of the journals, *exploration* and *diversity* are used more frequently than *exploitation* and *intensity*, with the *IEEE Transactions on Evolutionary Computation* mentioning *diversification* in more than eight times the number of papers than for *intensification*. Considering that a metaheuristic requires a balance between exploration and exploitation (as well as between diversity and intensity, as explained in Section 4.3), it is striking that the use of the terminology is not more balanced. This may point to a systematic bias toward exploration, resulting in under-performing algorithms.

Although the terms are ubiquitously employed, they do not have universally accepted definitions. Eiben and Schippers [54] reviewed how *exploration* and *exploitation* are used in the literature on EAs. They remark that "most authors leave their definitions implicit and use the intuitive

meaning of the concepts to explain the working of EAs" and "that there is no general consensus on these matters; several authors represent contradicting views."

However, they do acknowledge a few prevalent views: namely that "selection is commonly seen as the source of exploitation, while exploration is attributed to the operators mutation and recombination", "exploitation is the usage of information" and "that exploration and exploitation are opposite forces" which must be balanced. A sample of more recent papers confirms the continued expression of the first [24, 32, 78, 127, 171], second [32, 120, 185, 187, 188] and third views [1, 2, 32, 78, 127, 104, 132, 169]. Other prevalent views in the literature, not identified by Eiben and Schippers, include: "a latent viewpoint [which] interprets exploration and exploitation as global search and local search, respectively" [32] (also [95, 186, 189]), that exploitation is short-term whereas exploration is long-term [30, 31, 40], and that exploitation and exploration correspond to intensification and diversification, respectively [2, 22, 78, 104, 126, 129]. In summary, the prevalent views propound that exploitation and exploration are, respectively:

1. local and global search,
2. selection and reproduction operators,
3. information utilization and information acquisition,
4. short-term and long-term strategies,
5. intensification and diversification, and
6. opposite forces which must be balanced.

These are by no means the only views on the terminology. Many of the alternative views are less elucidatory, such as the circular: "*Exploitation* is the property of the algorithm to thoroughly *explore* a specific region of the search space, looking for any improvement in the best currently available solution(s). *Exploration* is the property to explore wide portions of the search space, looking for promising regions, where *exploitation* procedures should be employed" [119] (italics added). Another example of an uninstructive definition is, "Exploitation is defined ... as the ability of an algorithm to step into the direction of desired improvement" [16]. Stepping in the desired direction of improvement is the objective of most algorithms and is not specific to exploitation. Although there is some truth to these views, without further context or content they provide little value.

A recent study [41] confirms the lack of progress made in understanding the terminology. If anything, as the research community has grown, the situation has become worse as the number of views has increased with little effort being spent on separating the wheat from the chaff. Extracting the prevalent views from the literature and determining which are meaningful and consistent with each other is essential to clarifying the terminology. Without this the research community will remain unable to communicate its ideas effectively.

Demonstrating that each of the prevalent views on exploitation and exploration follow from fundamental definitions of these terms will unify, and thereby clarify, the terminology. The notion of a PFL is introduced in the following section, which is used as a basis for such definitions. In Section 4.3 it is demonstrated how each of the prevalent views may be deduced from the PFL.

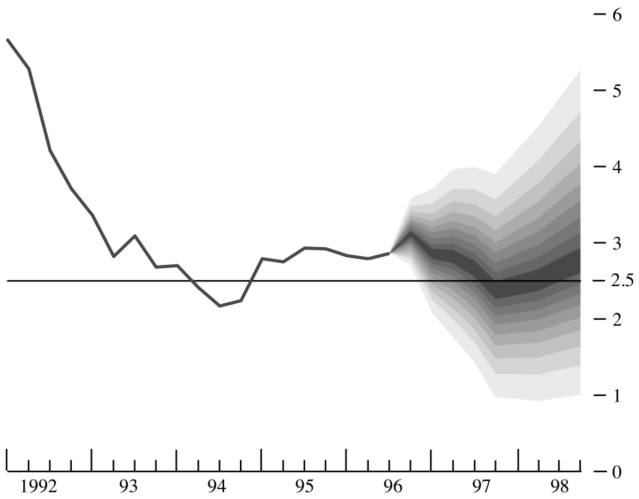


FIGURE 4.1: *Fan chart of inflation in Britain [10]. Observed past data are connected by a simple line chart until a certain time (1996), after which possible outcomes are projected. The dark band of future outcomes indicates the expected outcome, whereas less probable outcomes are displayed in lighter shades.*

4.2 The Probable Fitness Landscape

The notion of the PFL was inspired by that of a *fan chart*. Fan charts have been used since 1997 by the Bank of England to graphically describe its best prevision of future inflation [10]. The observed past data on inflation is connected by a simple line chart which diverges for future time values to represent a range of possible outcomes, with more probable outcomes having a darker shade of colour. An example of a fan chart is provided in Figure 4.1.

The outcome for all future time values is uncertain. Although for each time value there exists an outcome that could be calculated in the future, it is not currently known and may therefore be considered as random. This randomness is not complete, but defined by a *probability distribution* that depends on the information of the past observational data and known properties of inflation. Hence, for each future point in time there is a probability distribution of outcomes, as displayed in the fan chart.

This same notion may be applied to metaheuristics. Consider a continuous fitness function $f : \mathcal{S} \mapsto \mathbb{R}$, where $\mathcal{S} \subset \mathbb{R}^n$ is the search space of the optimisation problem. It is assumed, without loss of generality, that maximising the fitness function is the objective of the optimisation problem. The *Fitness Landscape* (FL) is the surface in $\mathcal{S} \times \mathbb{R}$ defined by the fitness function $(s, f(s))$, where $s \in \mathcal{S}$.

At any stage during the execution of a metaheuristic the current population (possibly consisting of a single candidate solution) is known, which is equivalent to past observational data. The known properties of the fitness function are analogous to the known properties of inflation. Together these may be used to construct the PFL, which graphically represents the fitness probability distribution of every point in the search space. In the PFL every coordinate in $\mathcal{S} \times \mathbb{R}$ is assigned a value according to a grey colour scheme. The darker a coordinate is, the larger the probability that it is in the FL.

If a candidate solution s_p is in the current population, then its fitness $f(s_p)$ is known with certainty. It is represented in the PFL by a black dot at $(s_p, f(s_p))$ with white for all coordinates in $\{s_p\} \times \mathbb{R} \backslash f(s_p)$.

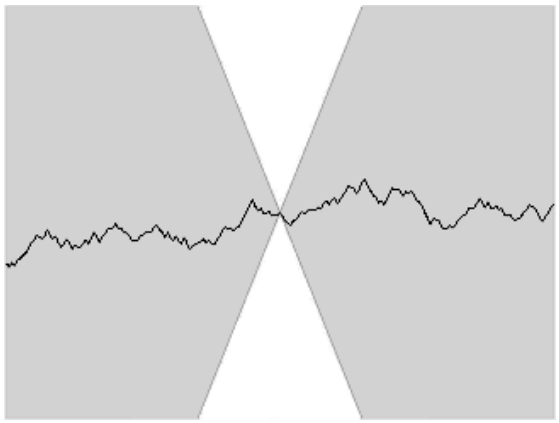


FIGURE 4.2: *A Lipschitz continuous function has the defining property that there exists a constant* $K \geq 0$ *such that* $|f(x_1) - f(x_2)| \leq K||x_1 - x_2||$ *for all* x_1, x_2*. This may be represented as a double cone (shown in white) whose vertex can be translated along the search space, so that the fitness function (shown in black) always remains entirely outside the cone [177].*

The fitness of all points not in the current population is uncertain. Although for each point there exists a fitness value that could be calculated, it is not currently known and may therefore be considered as random. This randomness is not complete, but defined by a *fitness probability distribution* that depends on the information of the current population and known properties of the fitness function. Hence, for each point in the search space there is a fitness probability distribution, as illustrated in the PFL.

If a point is not in the current population, then the fitness probability distribution cannot be determined with certainty, since this would require knowing the point's fitness value, which is uncertain. This reduces the PFL to a hypothetical construct that cannot be calculated with certainty. However, the PFL does have two governing principles. Firstly, s_p is of higher than average fitness, since it has survived selection. Therefore the *expected fitness* decreases away from s_p. Secondly, the further away a point in $\mathcal{S}$ is to s_p the larger the range of its possible fitness values, due to the (Lipshitz) continuity of the fitness function. The increase in range of possible fitness values according to Lipschitz continuity may be seen in Figure 4.2 (taken from [177]). Thus the *variance* of the fitness probability distribution increases away from s_p.

Figure 4.3 contains a graphical illustration of an example of the PFL for a population containing a single candidate solution. A single black dot is located at $(s_p, f(s_p))$, indicating that the point is definitely in the FL, whereas there is a fitness probability distribution for all other points in the search space. It is clear from the figure that the expected fitness decreases, whereas the variance of the fitness probability distributions increases, away from the candidate solution.

The notion of a PFL may be extended to populations with multiple candidate solutions, as shown in Figure 4.4. It is evident that the FL agrees with PFL at points in the current population, that is they both have single black dots corresponding to the fitness values of the points in the current population. For points not in the current population the PFL provides a good approximation of the FL, with darker points in the PFL having a higher probability of coinciding with the FL.

Note that the PFL of a population of multiple individuals is not the average of each individual's PFL. This is because in the PFL of one individual, the points of other individuals are of uncertain fitness. Hence, if averaged over all individual PFLs, the points of individuals in the current population would have uncertain fitness (which is false).

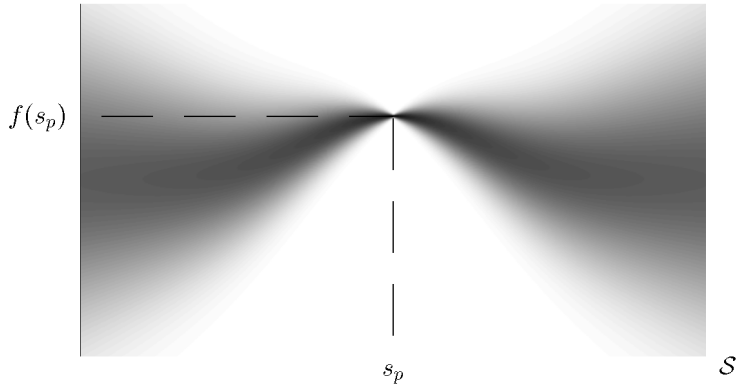


FIGURE 4.3: *The PFL of a population containing only one individual, s_p.*

A more accurate PFL can be constructed if more properties about the particular fitness function are known. In fact, the same current population may produce different PFLs, depending on the known properties of the fitness function. For example, a PFL associated with a rough fitness function will exhibit fitness probability distributions with more variance than that associated with a smooth fitness function. If metaheuristic has memory structures, such as a Tabu Search, then many features of the fitness function may be known and the fitness probability distributions might be very accurate. In fact, if the fitness values of points in previous populations have been recorded, then the fitness values of some points not in the current population are known with certainty and are represented by black dots in the PFL.

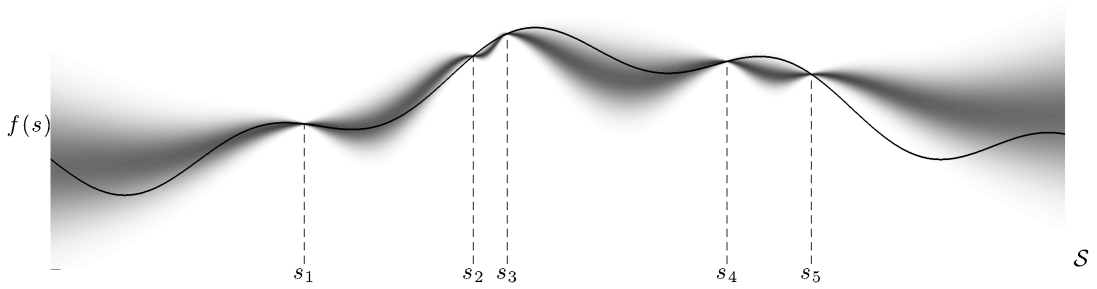


FIGURE 4.4: *The PFL of multiple individuals. A possible FL is shown as a solid line, whereas the PFL is a grey distribution.*

4.2.1 Meta-models

Even though the PFL cannot be constructed with certainty, there are techniques to approximate the FL known as *meta-models* (also called *surrogate models* or *fitness approximations*) [55, 91, 170]. Meta-models are typically used to estimate fitness values (equivalent to determining the expected fitness) of new candidate solutions if the computation of the actual fitness values is extremely time-consuming. This is done by interpolating the fitness values of all previously generated candidate solutions. One of the most popular meta-models is *Kriging* [98], for which the error estimation of the approximation (similar to the variance of the fitness probability distribution) may also be determined.

Although meta-models are very similar to the PFL, there are two differences. Firstly, the PFL only depends on the information of the current population and known properties of the fitness function, whereas meta-models usually use the information from all previously generated candidate solutions. Secondly, and more significantly, meta-models typically do not take into account the first principle of the PFL (that a candidate solution in the current population is of higher than average fitness). Thus, meta-models are not applicable when the current population only has one candidate solution and there is no record of the fitness values of previous points, as is the case for Simulated Annealing, since there are not enough points to interpolate. The PFL, on the other hand, is applicable under all conditions. In the case of Simulated Annealing the PFL would look similar to Figure 4.3, with the expected fitness away from the current candidate solution decreasing more sharply as the search progresses (due to the increase in the expected difference between the fitness of the current candidate solution and the average).

4.2.2 The history of the PFL

The PFL is built upon two notions, that of the fan chart, discussed at the beginning of this section, and the FL. The FL was first introduced by Sewall Wright, who is regarded as one of the three founding fathers of theoretical population genetics [149]. In 1931 Wright published his *Evolution in Mendelian Populations* [182], a highly mathematical paper for biologists, in which he proposed his shifting balance theory. A year later he was invited to give a very short presentation at the Sixth International Congress of Genetics [183] and, in order to condense his theory into an accessible form, he presented it pictorially using the metaphor of an *adaptive (fitness) landscape*. The landscape illustrated "the entire field of possible gene combinations [graded] with respect to adaptive value under a particular set of conditions," *i.e.* the fitness of every possible individual. He gave an example of such a landscape, reproduced in Figure 4.5.

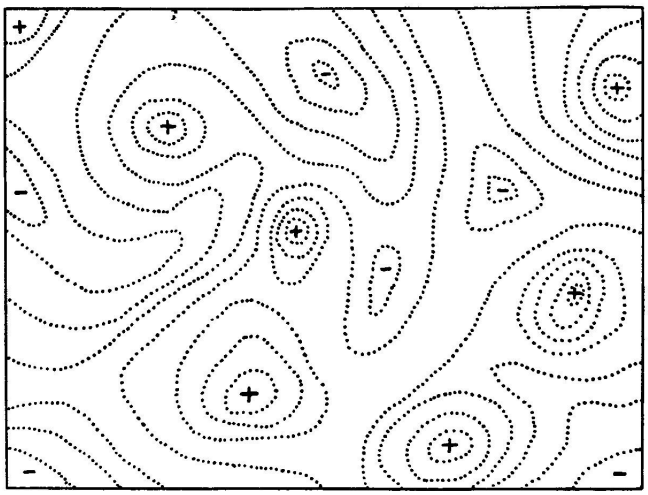


FIGURE 4.5: *Wright's original caption was: "Diagrammatic representation of the field of gene combinations in two dimensions, instead of many thousands. Dotted lines represent contours with respect to adaptiveness" [183]. Translated into EA language, it is a two-dimensional representation of the fitness of points in the search space, with hills signified by + symbols and valleys by − symbols.*

The metaphor proved to be hugely successful and influential — it came to be offered as the crucial key to the understanding of evolution [149]. Although the landscape did not contain any

more information than the formal mathematics of Wright's paper, it presented the information in a readily understandable format which catalysed its adoption.

Many decades later the EA community co-opted the FL as a method of explaining algorithm performance [141]. For instance, Figure 2.3 in Chapter 2 uses the FL to illustrate the three escape strategies for local maxima. The PFL seeks to refine the metaphor of the FL for EAs by incorporating only the information known during the execution of an EA, not the fitness of every possible individual. Since the PFL represents the actual situation of the EA practitioner, it is more appropriate than the FL. However this does not make the PFL a replacement for the FL, but rather a complementary notion that can be used together with the FL to analyse EAs.

Arguably the PFL is, and has always been, the fundamental concept that researchers have implicitly used to devise new EAs. In fact, perhaps the first paper ever written on metaheuristics, entitled *A stochastic approximation method* [143] and published in 1951, is built on an idea similar to the PFL. The first section of the paper reads,

> "Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant."

It goes on to

> "suppose that to each value x corresponds a random variable $Y = Y(x)$ with distribution function $\Pr[Y(x) < y] = H(y|x)$, such that
>
> $$M(x) = \int_{-\infty}^{\infty} y \, \mathrm{d}H(y|x)$$
>
> is the expected value of Y for the given x."

The function $M(x)$ is equivalent to the fitness function $f(x)$ and the variable $Y(x)$ is like the range of possible fitness values of point x in the PFL[1].

It is clear that researchers have been thinking about expected values and distribution functions from the very beginning. The advantage of formalising these notions in the form of the PFL, with its two governing principles, is twofold. Firstly, the governing principles may be used to formally deduce the consequences of the concept: to define relevant terminologies and explain performance, as the fan chart does in banking. And secondly, the PFL illustrations are a catalyst for understanding EAs, playing the same role as the FL does in evolutionary biology. Recently the PFL has also taken on a third role of analysing landscapes and designing algorithms, as explained below.

4.2.3 Practical application — MAX-3-SAT

In 2012 Prügel-Bennett and Tayarani-Najaran published a paper [137] in which they analyse the MAX-3-SAT problem. This problem is closely related to the satisfiability decision problem colloquially known as 3-SAT, which involves a set of Boolean variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and a set of disjunctive clauses consisting of 3 literals[2]. For example, a set of clauses might be

[1] Here the similarity between the paper and the notion of a PFL ends. The paper continues to propose a metaheuristic that is proven to converge in probability.

[2] A literal is either a variable or its negation.

$\{X_1 \vee \neg X_5 \vee X_{10},\ X_2 \vee X_4 \vee \neg X_8,\ X_1 \vee X_7 \vee X_9\}$. In 3-SAT the question is, "Does there exist an assignment of the variables which satisfies all the clauses?" [137].

MAX-3-SAT is the generalization of 3-SAT to problems which are not fully satisfiable. It asks the question, "Given m clauses, what is the maximum number of them that can be satisfied by an assignment?" MAX-3-SAT may thus be formulated as an optimization problem where the objective function is the number of satisfied clauses. Assuming there are m clauses and denoting the clauses by $g_i(\mathbf{X})$, then the fitness is given by

$$f(\mathbf{X}) = \sum_{i=1}^{m} [g_i(\mathbf{X}) \text{ is satisfied}]$$

where $[g_i(\mathbf{X})$ is satisfied$]$ is an indicator function equal to 1 if clause g_i is satisfied and 0 otherwise [137].

Although the problem is discrete, in the paper it is argued that "the landscape is relatively smooth" and that "the configurations around a fit configuration also tend to be fit." This suggests that there is some form of continuity in the landscape, a claim that can be substantiated through the PFL. If the landscape expresses the characteristics of the PFL, so the argument goes, it must fulfil the requirement of the PFL, namely, that the landscape is continuous.

Using the properties of MAX-3-SAT the expected fitness and variance are analytically determined in a Hamming sphere around any configuration, the results of which are displayed in Figures 4.6 and 4.7. It is clear that the expected fitness decreases whereas the variance increases away from a local maximum, confirming the two principles of the PFL. Figure 4.7 also illustrates that the fitness around fit individuals must decrease faster than for unfit individuals, as there is a greater difference between the fitness of the individual and the average fitness of the search space. The authors conclude that there exist long-range correlations in the landscape (a stronger conclusion than continuity) and develop an algorithm which utilizes this property. Although the conclusion is debatable, this does demonstrate how the properties of the PFL may be used to analyse landscapes and design algorithms.

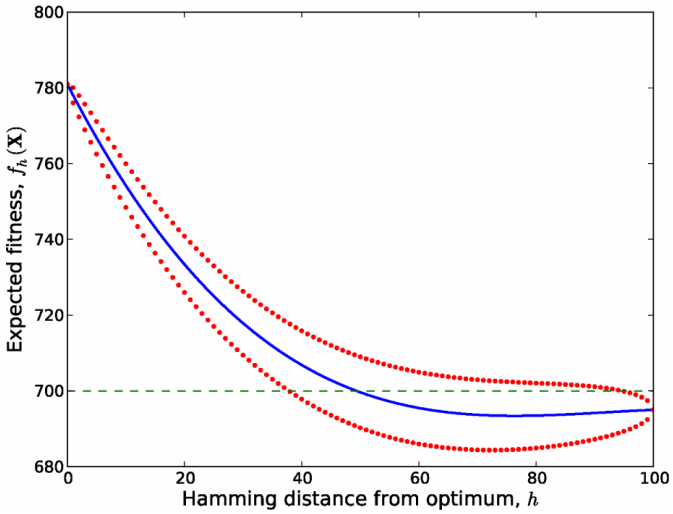


FIGURE 4.6: *Expected fitness of configuations in a Hamming sphere of radius h around a local maximum. The dotted curves show one standard deviation around the mean. The average fitness in the search space is shown by the horizontal line. [137]*

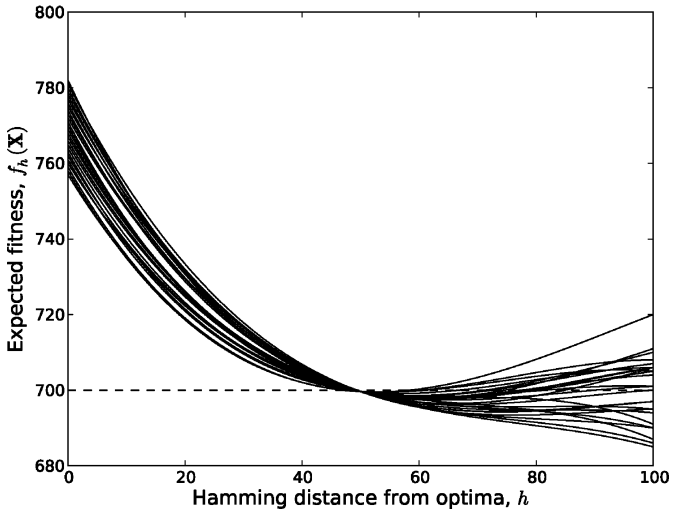


FIGURE 4.7: *Expected fitness in a Hamming sphere around a number of local maxima, each represented by a separate curve. [137]*

The rest of the chapter focusses on the theoretical aspects of the PFL, initially in defining terminologies and then in explaining operator performance.

4.3 Unification of prevalent views

At each iteration during the execution of a metaheuristic, new candidate solutions must be generated. From the PFL definitions of *exploitation* and *exploration* can be made as follows:

> **Exploit:** to generate candidate solutions at points of high expected fitness,
>
> **Explore:** to generate candidate solutions at points of high variance.

These novel definitions come directly from the notions of expected fitness and variance in the PFL and therefore share the same limitations as the PFL. Since the PFL cannot be explicitly calculated, it is impossible to determine the exact expected fitness of a point. Hence it is impossible to determine the degree of exploitation or exploration in generating a candidate solution at that point. However, it can be used to compare two points. If a point is closer to a candidate solution, then it has a higher expected fitness and generating a candidate solution at that point is more exploitative. Likewise, the closer point has lower variance and generating a candidate solution at that point is less explorative.

The prevalent views on exploitation and exploration can be deduced from the above definitions of exploitation and exploration. If the views stem from the same definitions, then they are necessarily consistent and simply represent different insights into the same phenomenon.

4.3.1 Local and global search

Local and global search are not themselves well defined terms. It is understood that local search refers to a search which is only able to reach a local maximum, whereas a global search may find

any maximum. They may be thought of as hill climbing and pure random search, respectively (see Chapter 2). The ability to generate a candidate solution at any point in the search space, including the global maximum, is the key characteristic of a global search. By contrast, a local search is unable to generate points outside of the neighbourhood of a local maximum. Hence the essential characteristic of local search is that it only generates close to a candidate solution, whereas global search may generate far from a candidate solution.

Exploitation generates new candidate solutions at points of high expected fitness. These points are close to candidate solutions and therefore correspond to a local search. Meanwhile exploration generates at points of high variance, which are far away from current candidate solutions, corresponding to a global search.

4.3.2 Selection and reproduction operators

Exploitation and exploration have been used to refer to operators of an algorithm. An example is embodied in the following definitions [24]:

> **"Exploitation** indicates the parts of an EA that are concerned with the selection of a set of parent solutions from the current population and the construction of a new population given the current population, the selected set of parent solutions and the set of offspring solutions. This definition of exploitation thus includes traditional selection, but also all replacement schemes such as crowding."
>
> **"Exploration** indicates the part of an EA that is concerned with the generation of new offspring solutions from a given set of parent solutions..."

This agrees with the prevalent view that "selection is ... the source of exploitation, while exploration is attributed to the operators mutation and recombination."

To see how this view follows from the above definitions, the principles of the PFL are appealed to. Without selection the first governing principle of the PFL, namely that a candidate solution in the current population is of higher than average fitness, would fail. Hence exploitation, which generates at points of high expected fitness, would be impossible. Thus selection is the source of exploitation. Exploration, which generates at points of high variance, is possible without selection. This is because the second governing principle of the PFL, namely that the further away a point in $\mathcal{S}$ is to s_p the larger the range of its possible fitness values, would still hold true. Hence operators which explore, *i.e.* generate candidate solutions at points of high variance, would still be possible, and so exploration is attributed to the operators mutation and recombination.

4.3.3 Information utilization and information acquisition

According to Chen *et al.* [32], "in learning algorithms, exploration and exploitation correspond to the acquisition and utilization of knowledge, respectively." Two propositions are bound together in this claim. The first is that exploitation, opposed to exploration, utilizes knowledge and the second is that exploration, opposed to exploitation, acquires knowledge. Both statements are true in part, but not exclusively so.

The expected fitness of a point depends on both the distance from and the fitness values of the candidate solutions in the current generation, whereas the variance just depends on the distance. Hence exploitation (which depends on expected fitness) utilizes more information

than exploration (which in turn depends on the variance). Both do utilize information, just more so for exploitation as opposed to exploration.

Whenever a candidate solution is generated at a point and its fitness is evaluated, information is acquired about the fitness function. Both exploitation and exploration involve generating candidate solutions; therefore both acquire information. The difference is that exploitation can only generate in a local neighbourhood, whereas exploration can generate anywhere in the search space (see Section 4.3.1). Hence exploration can gather more information, or at least a greater range of information, than exploitation can. Again, both do acquire information, just more so for exploration than for exploitation.

4.3.4 Short-term and long-term strategies

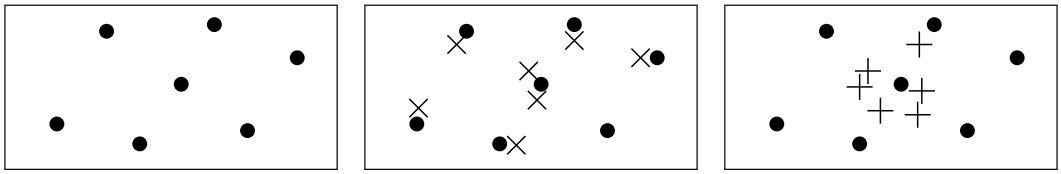
The issue of short-term and long-term strategies is connected to that of information utilization and information acquisition. The utilization of information is of short-term benefit, while the acquisition of knowledge is advantageous in the long-term. Hence, exploitation, which utilizes information, is a short-term strategy; whereas exploration, which acquires knowledge, is a long-term strategy.

A consequence of this is that exploration should be favoured toward the beginning of a search, when there are many iterations left to benefit from a long-term strategy. On the other hand, exploitation should be prioritised at the end, since at that stage a short-term strategy will be more fruitful.

4.3.5 Intensification and diversification

The terms *intensification* and *diversification* are often used interchangeably with exploitation and exploration, respectively [2, 126]. However, they have subtly different meanings, as noted by Blum and Roli [22, p.271], "The term diversification generally refers to the exploration of the search space, whereas the term intensification refers to the exploitation of the accumulated search experience. These terms [(diversification and intensification)] stem from the Tabu Search field and it is important to clarify that the terms exploration and exploitation are sometimes used instead, for example in the Evolutionary Computation field, with a more restricted meaning."

The main difference is that exploitation and exploration refer to points in the search space, whereas intensity and diversity refer to the distribution of candidate solutions in the search space. This distinction is evident in Figure 4.8. On the left there is a plot of the initial population with possible future generations displayed in the centre, by crosses, and on the right, by plus symbols. The cross population exhibits almost no exploration yet maintains a diverse population, while the plus population is highly explorative, but results in an intense distribution.



(a) *Initial Population* (b) *Diversity without exploration* (c) *Intensity with exploration*

FIGURE 4.8: *Populations exhibiting diversity without exploration, and intensity with exploration.*

The examples in Figure 4.8 are atypical. Exploration generates far away from candidate solutions which tends to create a diverse population, whereas exploitation generates close to candidate solutions, generally resulting in an intense population. Hence the terms *diversification* and *intensification*, meaning *the process of making diverse* and *the process of making intense*, respectively, are sometimes used interchangeably with exploration and exploitation.

These notions may be extended beyond the current population to the set of all candidate solutions that have been generated throughout the search. This makes the terms applicable to metaheuristics, such as Tabu Search, that have populations consisting of only one candidate solution and memory structures. The terminology has evolved to reflect the different meanings, with intensification and diversification traditionally referring to the set of all candidate solutions, whereas exploitation and exploration refer to the current population. Blum and Roli confirm this usage by stating that "exploitation and exploration often refer to rather short-term strategies tied to randomness, whereas intensification and diversification also refer to medium- and long-term strategies based on the usage of memory" [22, p.271].

4.3.6 Opposite forces which must be balanced

All of the above prevalent views have contrasted exploitation and exploration as opposite forces. This ultimately stems from the PFL where the expected fitness decreases away from candidate solutions, whereas the variance of the probability distributions increase. Hence exploitation and exploration have opposing tendencies, to generate close to and far from candidate solutions, respectively.

However, exploitation and exploration are not direct opposites. There may be points of high expected fitness and high variance (a moderate distance away from a candidate solution with an extremely high fitness value), or low expected fitness and low variance (close to a candidate solution with a very low fitness value).

Since exploitation and exploration are opposite forces, both of them can be controlled by the same operator. For instance, even though in Section 4.3.2 selection is argued to be the source of exploitation, it may also be used to maintain or enhance exploration via niching, preselection or fitness sharing [112, 113]. Likewise, some reproduction operators, such as crossover, may affect exploitation.

The reason for balancing exploitation and exploration is evident from considering their extreme forms. Extreme exploitation is simple hill climbing, unable to escape the region of a local maximum, while extreme exploration is pure random search, incapable of incremental iterative improvement (see Chapter 2). Neither of these extremes are ideal and instead a combination is required for a successful search. It is clear that both exploitation and exploration have their own unique benefits, as examined in the following section.

4.4 The benefits of exploitation, exploration and diversity

The benefit of exploitation is obvious — the objective of the problem is to generate points of high fitness! Below it is demonstrated that it is also beneficial to explore and maintain a diverse population.

4.4.1 The benefit of exploration

Imagine a red bag full of red balls and a blue bag full of blue balls. The red balls have the monetary values \$0 and \$2, whereas blue balls have the values −\$3 and \$3 (with balls of each value being in equal proportion). It is clear that the red balls have the higher average value, but exhibit lower variance than the blue balls. Balls are drawn n times from each of the bags, after which each bag is assigned the value of its highest value ball drawn. The problem is to choose one bag such that your profit is maximised.

For each bag the probability that the higher value ball is drawn at least once after n draws is $1 - 2^{-n}$. Thus, the expected value for the red bag is $2 - 2^{1-n}$ dollars, whereas the expected value for the blue bag is $3 - 3 \cdot 2^{1-n}$ dollars. Since the value of the blue bag will either be lower $(-3 < 0, 2)$ or higher $(3 > 0, 2)$ than whatever the value of the red bag is, there is probability of $1 - 2^{-n}$ that the blue bag will yield the greater profit.

The best bag to choose can be made according to the following argument:

- For $n = 1$ both bags have an equal probability of yielding the highest value and the red bag has a higher expected value. Therefore, it would be reasonable to draw from the red bag.
- For $n = 2$ the blue bag has a greater probability of yielding the higher value and both bags have the same expected value. Therefore, it would be reasonable to draw from the blue bag.
- For $n \geq 3$ the blue has both a greater probability of yielding the higher value and a higher expected value. Therefore, it would be reasonable to draw from the blue bag.

This demonstrates that a high expected value (exploitation) is of greater concern in the short run; but high variance (exploration) is more important in the long-term. This agrees with the findings in Section 4.3.4.

4.4.2 The benefit of diversity

Now consider the situation where the values for both colour balls are −\$1 and \$1 (again in equal proportion). The red balls now also have the property that the value of the balls drawn is always the same, *i.e.* without diversity. That is, if a red ball of value −\$1 is initially drawn then all subsequent red balls will have the value −\$1; and likewise if a red ball of value \$1 is initially drawn then all subsequent red balls will have the value \$1. The values of the blue balls are independent, with each ball drawn having an equal probability of being −\$1 or \$1, resulting in a diversity of possible values. The expected value for the red bag is 0 dollars, while for the blue bag it is $1 - 2^{1-n}$ dollars; and the probability of the blue bag having the higher value is $1 - 2^{-n}$. For $n \geq 2$ the blue bag has both a greater probability of having the higher value and a higher expected value. Therefore, it would be reasonable to draw from the blue bag and it may be concluded that diversity is beneficial.

4.5 Utility and the IPD

Throughout the execution of an EA a combination of exploitation and exploration is required, although there are occasions where the one should be prioritised over the other. For example,

toward the end of a search the short-term strategy of exploitation is preferable. Alternatively, if the population prematurely converges then local search becomes fruitless and the global search of exploration is more suitable. Inevitably the priority given to exploitation and exploration will vary as the search progresses, depending on the current population and computational constraints.

The relative priority of exploitation and exploration specifies the *utility* of high expected fitness and high variance. The PFL represents the expected fitness and variance of each point in the search space. Combining the relative priority of exploitation and exploration with the PFL, the utility of each point in the search space may be determined.

In general, points of higher expected fitness (assuming equal variance) or higher variance (assuming equal expected fitness) are of greater utility. The difficulties arise when comparing points without equal variance or equal expected fitness, in which case the relative utility depends on whether exploitation or exploration is being prioritised. For instance, a point of low expected fitness and high variance is of great utility if exploration is prioritised, but of little utility if exploitation is prioritised.

A *Probability Distribution (PD)* is a function $P : \mathcal{S} \mapsto [0, 1]$, where $P(s)$ is the probability that a candidate solution will be created at point s during the next generation. In most algorithms the PD is implicit, bound up in the EA operators (with an exception being Estimation of Distribution Algorithms [100, 168]). The *Ideal Population Distribution (IPD)* may be thought of as the ideal PD from which the next generation of individuals should be produced. It is a PD for which the probability is perfectly proportional to the utility of a point. Unfortunately, the IPD cannot be explicitly calculated since it is a hypothetical construct, based on another hypothetical construct, the PFL.

4.6 Analysis of EA operators

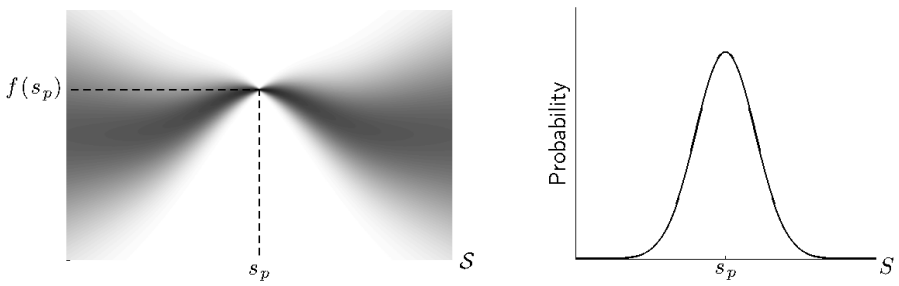
During each generation the EA operators specify a PD from which individuals are generated. The effectiveness of these operators may be judged on three criteria: *the agreement of the PD with the IPD*, *population diversity* and *computational speed*. In the subsequent subsections the principle EA operators, mutation and selection, are evaluated according to these criteria.

4.6.1 Mutation

Mutation is the simplest EA operator for generating children from parents. During an iteration each parent is mutated from its current position according to a PD in order to generate a child.

Consider a population consisting of only one individual, in which case the PD of the algorithm is that of the one parent. Typical mutation PDs, for instance a Gaussian distribution (see Section 3.1.1), are symmetric, unimodal and centred at the origin. Such a PD agrees with the IPD since the distribution is roughly proportional to the expected fitness, as is clear from Figure 4.9. It is largely this agreement that makes mutation a successful operator.

If multiple individuals are present in a population then the algorithm's PD is the average of the parents' PDs, as shown in Figure 4.10. However, each child is independently generated from a parent's PD, not from the algorithm's PD. This affects the agreement of the algorithm's PD with the IPD. To see this, consider a point where the parent's PDs overlap (significantly). The value of the algorithm's PD at this point is based on the average of the parents' PDs, which in turn is based on the average of the parents' PFLs. As noted in Section 4.2, the PFL of a population



(a) *The PFL* (b) *The PD*

FIGURE 4.9: *The PFL and PD of a population consisting of one individual* $s_p \in S_p$.

of multiple individuals is not the average of each individual's PFL. Hence the algorithm's PD at this point is not accurate, since it is based on the average of each individual's PFL (not the actual PFL of the population). A consequence of this is that if the mutation PD's variance is too large, then the PDs of individuals tend to overlap and the algorithm's PD does not reflect the IPD.

Another effect of overlapping PDs is that diversity is diminished. To demonstrate this, consider two parents that are far apart (such that their PDs do not significantly overlap). If each parent generates one child, then the children are likely to be far apart and the population is therefore diverse. Alternatively, if the averaged PD is drawn from to generate the children, then half of the time both children are drawn from the same parent's PD. In this case it is likely that the children are close together and the population is less diverse.

The final reason for mutation's success is its computation speed — in general, mutation is far faster than recombination operators. Together with the PD's good agreement with the IPD and the promotion of population diversity, mutation scores well on all three criteria for judging operators. This explains the effectiveness of mutation and, consequentially, why it is a standard feature of EAs.

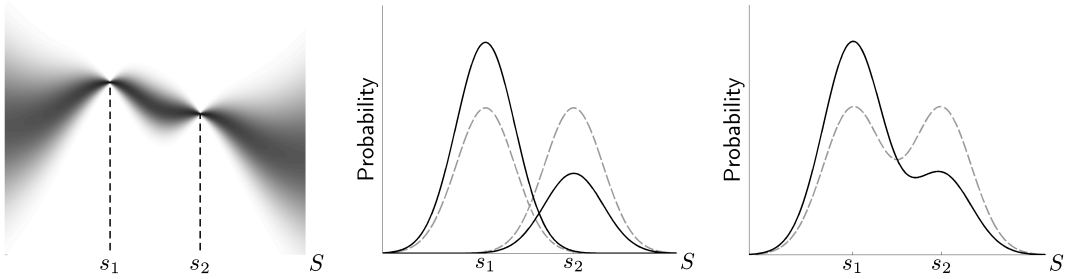
4.6.2 Selection

The purpose of selection becomes clear when considering multiple individuals of different fitness values. Close to fit individuals there is high expected fitness and low variance, whereas close to unfit individuals there is low expected fitness and low variance. Points of higher expected fitness, with equal variance, are of greater utility. Thus the IPD around a fit individual exhibits a larger probability than around an unfit individual.

Selection increases the probability of generating children around fit parents. This is achieved by probabilistically duplicating fit parents and deleting unfit parents. Once selection has taken place, each parent is mutated as usual. As may be seen in Figure 4.10, mutation with selection produces a PD which is in better agreement with the IPD.

Selection necessarily involves not generating children from unfit parents, which decreases the number of parents that are generated from. Since multiple children are drawn from the same parent's PD, the children are located closer together and the population is less diverse.

Thus selection causes an algorithm's PD to be in greater agreement with the IPD, but at the cost of decreasing population diversity. Fortunately the cost can be countered by making mutation



(a) *The PFL* (b) *The PDs of each individual* (c) *The combined PDs*

FIGURE 4.10: *The PFL, separate PDs and an algorithm's PD of a population consisting of two individuals $s_1, s_2 \in S_p$ with different fitness values. The solid black line is with selection and the dashed grey line is without.*

create a more diverse population (*e.g.* by increasing the mutation PD's variance).

4.7 Chapter summary

This chapter is the most crucial in the thesis, since it introduces the notion of the PFL. The PFL was developed in response to the literature review presented at the beginning of the chapter, which raised a few issues. Firstly, that the notions of exploitation, exploration, intensity and diversity are critical to the understanding EAs, as they appear in a large majority of prominent academic journals related to EAs. Secondly, the terms exploration and diversity are used far more frequently than exploitation and intensity, which may point to a systematic bias toward exploration that results in under-performing algorithms. And thirdly, there are a number of prevalent views on exploitation and exploration in the literature. It was not initially clear whether these views were meaningful or consistent. To investigate this, six prevalent views were identified. These propounded that exploitation and exploration are, respectively: local and global search, selection and reproduction operators, information utilization and information acquisition, short-term and long-term strategies, intensification and diversification and opposite forces which must be balanced. The question as to whether these views are meaningful and consistent with one another was addressed using the novel notion of the PFL.

The PFL was inspired by the idea of a fan chart, used in banking, to graphically illustrate uncertain values. It does this by treating the uncertain values as random, with the randomness being defined by a PD that depends on the information of past observational data and other known properties. Applied to EAs, the PFL represents the PDs of the fitness values for each point in the search space with a grey colour scheme, examples of which may be seen in Figures 4.3 and 4.4. Since the PFL represents uncertain values, it cannot be calculated with certainty, although there are two principles of the PFL which give guidelines for comparing the fitness PDs of two points in the PFL.

Historically, it is evident that the idea of the PFL has been around since the beginning of EAs. However, there are new uses being developed. An example of this is the work by Prügel-Bennett and Tayarani-Najaran, which use the principles of the PFL to infer some sort of continuity (discussed in Section 4.2.3). A major contribution of this thesis is the novel use of the PFL as a basis for definitions of exploitation and exploration (see Section 4.3). From these definitions, the six identified prevalent views on exploitation and exploration may be deduced. This shows that all of the views are meaningful and consistent, answering the question posed by the literature

review, and the terminology may therefore be used in research effectively.

The PFL may be further extended to the notion of the IPD, which is a hypothetical ideal PD for generating new candidate solutions. This may be used, in combination with diversity and computational speed considerations, to determine why certain operators are effective. The respective purposes of mutation and selection are explained using this type of "PFL argument" in the penultimate section of the chapter. The same style of PFL analysis may be applied to any EA operator. Although it does not prove or even empirically substantiate an operator's performance, it does offer a qualitative argument as to why an operator should perform well. In the subsequent chapters this analysis is applied to the unique operators of GAs, ESs and EP.

CHAPTER 5

Genetic Algorithms

Contents

The field of GAs was largely inspired by the work of John Holland. In 1975 he authored a groundbreaking book entitled *Adaptation in natural and artificial systems* [68] that laid the practical and theoretical foundation for GAs. Since then the algorithm's popularity has grown, spurred on during the late 1980s by a highly influential textbook by Goldberg [83], into perhaps the most widely used metaheuristic today.

Influenced by genetics, GAs traditionally employ binary solution encoding (analogous to DNA) and employ both mutation and crossover to generate successive populations of candidate solutions. The order in which the operators are applied is crossover, then mutation and then selection (which is conventionally fitness proportional — see Section 3.2.4). The reason why crossover is applied before mutation is so that the properties of the parents may be combined in order to create children with the best characteristics of both. Whether this is accomplished or not is discussed in the second half of the chapter, while crossover operator is introduced in the first half, accompanied by various analyses of its properties.

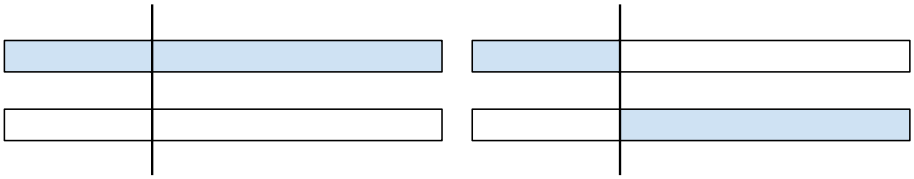
5.1 Crossover

Traditional crossover takes two parents from the current population to produce two children. The type of crossover first proposed by Holland [68] is called *one-point crossover*, in which bits

to the right of a chosen *crossover position* are exchanged between the individuals. For example, consider two parents $\mathbf{v} = (v_1, v_2, \ldots, v_k)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_k)$ with the crossover position ℓ (with $0 < \ell < k - 1$). The children produced by crossover are

$$\begin{aligned} \mathbf{v}' &= (v_1,\, v_2,\, \ldots, v_\ell, \quad w_{\ell+1}, w_{\ell+2}, \ldots w_k) \qquad \text{and} \\ \mathbf{w}' &= (w_1, w_2, \ldots, w_\ell, \quad v_{\ell+1},\, v_{\ell+2},\, \ldots v_k), \end{aligned}$$

which may be represented schematically as in Figure 5.1.



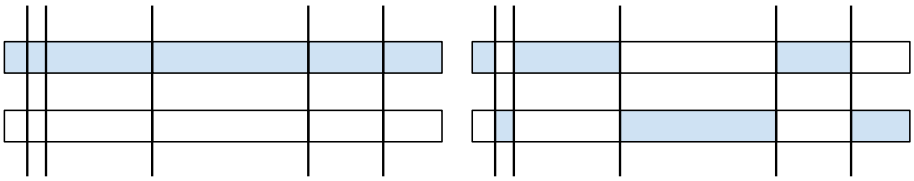
(a) *Parents* (b) *Children*

FIGURE 5.1: *Schematic representation of one-point crossover. The vertical line demarcates the crossover position.*

One-point crossover may be generalised into *multi-point crossover*. In this case not one but z crossover positions are chosen[1], $\ell_1, \ell_2, \ldots, \ell_z$, with $0 < \ell_i < \ell_{i+1} < k-1$ for all $i = 1, 2, \ldots, z-1$. Bits are exchanged every second segment between subsequent crossover positions to yield the children

$$\begin{aligned} \mathbf{v}' &= (v_1,\, v_2,\, \ldots, v_{l_1}, \quad w_{l_1+1}, w_{l_1+2}, \ldots, w_{l_2}, \quad v_{l_2+1},\, v_{l_2+2},\, \ldots) \qquad \text{and} \\ \mathbf{w}' &= (w_1, w_2, \ldots, w_{l_1}, \quad v_{l_1+1},\, v_{l_1+2},\, \ldots, v_{l_2}, \quad w_{l_2+1}, w_{l_2+2}, \ldots), \end{aligned}$$

which may be represented schematically as in Figure 5.2.



(a) *Parents* (b) *Children*

FIGURE 5.2: *Schematic representation of multi-point crossover. The vertical lines demarcate the crossover positions.*

The advantage of multi-point over one-point crossover is that it reduces *linkage* (also known as *positional bias*). Linkage refers to the dependence of the exchange probability on the bit positions, *i.e.* bits which are closer together have a higher probability of being exchanged together. In general the probability of two adjacent bits being exchanged together is $1 - z/(k-1)$ (with $z = 1$ for one-point crossover and $z > 1$ for multi-point crossover); thus the linkage decreases as z increases. *Uniform crossover* completely eliminates linkage by making the probability of exchanging bits independent of one another, with each bit having a probability p_u of being exchanged.

[1] If z is odd, then an additional crossover position at point k is added.

5.2 Properties of crossover

This section analyses various properties of crossover. First the distance between children is examined, then genetic drift is focussed on, after which the crossover PD is more thoroughly analysed and compared to mutation. The properties mentioned here reflect a small subset of the properties discussed in the literature and also includes some novel contributions.

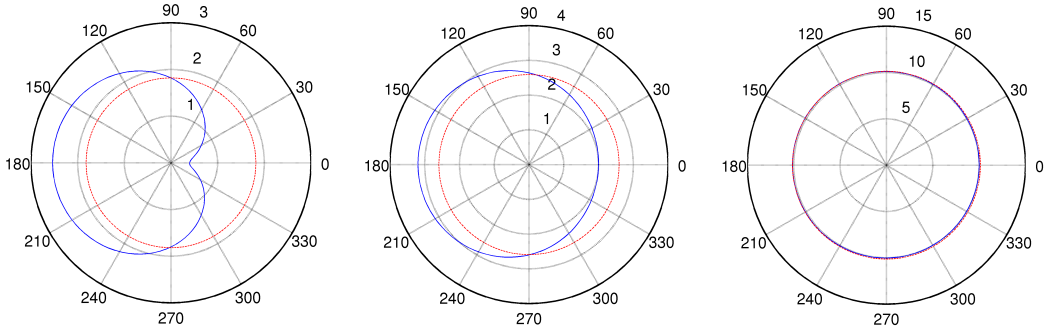
5.2.1 Relative position of individuals

Consider again two parents $\mathbf{v} = (v_1, v_2, \ldots, v_k)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_k)$ which undergo one-point crossover to generate two children $\mathbf{v}'$ and $\mathbf{w}'$. They may be represented as

$$v = \sum_{i=0}^{k} v_i 2^i, \quad w = \sum_{i=0}^{k} w_i 2^i, \quad v' = \sum_{i=0}^{l} v_i 2^i + \sum_{j=l+1}^{k} w_j 2^j \quad \text{and} \quad w' = \sum_{i=0}^{l} w_i 2^i + \sum_{j=l+1}^{k} v_j 2^j.$$

It is clear from the above expressions that $v + w = v' + w'$, in other words, the mean of the parents is equal to the mean of the children. This result may easily be generalised for multi-point and uniform crossover. An implication of this is that the mean of a population does not move due to generating new individuals via crossover, but by selection.

Another consequence of the mean being preserved is that the step vector of one individual is the exact opposite of the other, phrased mathematically as $v - v' = -(w - w')$. The 180^0 rotation of the step vector affects the expected distance between children, as novelly described below.



(a) *Parent separation of 0.2* (b) *Parent separation of 1* (c) *Parent separation of 5*

FIGURE 5.3: *Average distance between children with different initial parent separation distances. The step vectors are generated from a Gaussian distribution of variance one. Red represents independent step vectors and blue represents step vectors rotated at an angle.*

Contemplate the situation where two parents each create a child by adding a step vector to their original position[2]. There are two options: that the step vectors are independent (as in mutation), or that the angle between the step vectors is fixed. For both options the PD of the position of the children (averaged over the two parents) is the same, but the relative position of the children is different. These options were simulated for parents with different separation distances for which the average distance between children was calculated, with the results shown in Figure 5.3. It is clear from the figure that the average distance between the children is maximised if the angle between the step vectors is 180^0. The effect is more pronounced if the parents are closer together and disappears as the parents become further apart. Since crossover rotates the step vector by 180^0, it may be concluded that crossover leads to children being further apart than for mutation.

[2]It is assumed that both parents use the same rotationally invariant PD to generate step vectors.

5.2.2 Distance Correlation

There is a positive correlation between the distance separating parents and the step magnitude. This claim is substantiated by the results of applying crossover to all possible pairs of individuals of length seven, enumerated over all possible combinations and cut positions, as displayed in Figure 5.4. The correlation is weak due to the emergence of Hamming cliffs, although the line of best fit clearly indicates that the correlation is positive. It appears that 2-point and uniform crossover have similar step magnitudes, which are are roughly double that of 1-point crossover.

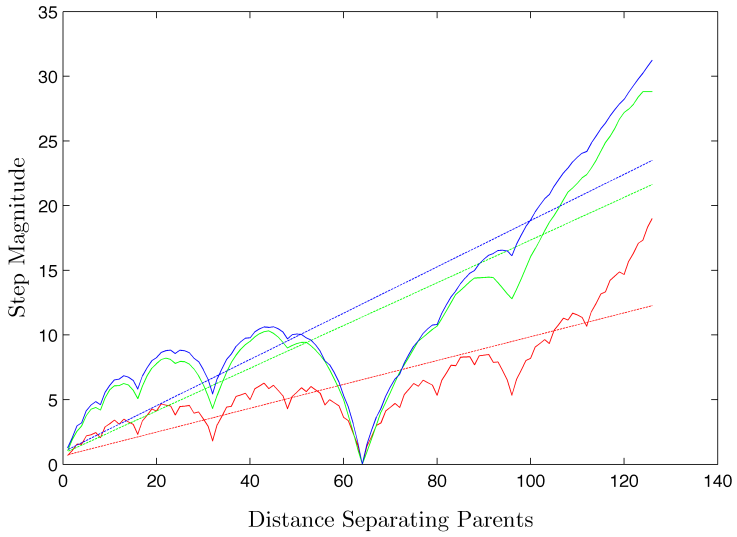


FIGURE 5.4: *Graph of the step magnitude against the distance separating parents. The colour blue represents uniform crossover, green 2-point crossover and red 1-point crossover. The straight lines depict the line of best fit.*

5.2.3 Genetic Drift

The distance correlation affects a phenomenon known as *genetic drift*. Genetic drift accounts for the random variation in populations not due to any influence of the fitness function. For example, two points A and B could be of equal fitness yet there may be more A individuals than B individuals in the population.

Consider a fitness function with two hills, phrased mathematically as $f(s) = 0$ if $1 < |s| < 2$ and $f = -\infty$ otherwise and shown graphically in Figure 5.5. Individuals of type A are defined as being points on the hill bound by -2 and -1, whereas individuals of type B are on the other hill, bound by 1 and 2. If an individual is generated outside of a hill then it has a fitness value of negative infinity and cannot be selected to participate in generating the next generation. Such an individual is replaced by either an A or B individual, with the respective probabilities proportional to the number of A or B individuals in the current population. Finally, suppose that the algorithm is steady state, with two parents undergoing crossover to produce two children, which replace their parents in the population.

Let m be the probability of a mixed pair (AB), and i be the probability of an identical pair (AA or BB), generating children outside of a hill during crossover, with $m > i$. Furthermore, assume that the probability of an identical pair generating children of a different type, or a

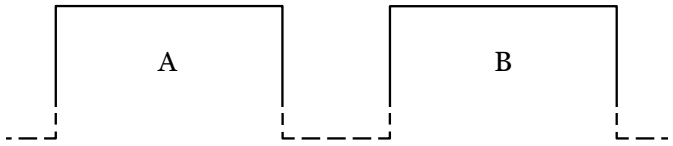


FIGURE 5.5: *The fitness function $f(s) = 0$ if $1 < |s| < 2$, or $-\infty$ otherwise. A individuals are situated on the left hill and B individuals on the right hill.*

mixed pair generating children of the same type, is zero. These assumptions are reasonable due to the correlation between step magnitude and distance separating parents as well as the 180^0 rotation of the step vector. The properties are summarised in Table 5.1.

Parent/Child	AA	BB	AB	Outside Hill
AA	$1-i$	0	0	i
BB	0	$1-i$	0	i
AB	0	0	$1-m$	m

TABLE 5.1: *The probablity of children being generated by pairs of parents via crossover. The rows represent the parents and the columns the children.*

According to the properties above, the iterative formula for the number of A individuals in generation $k+1$ for a population of size μ is

$$\begin{aligned} A_{k+1} &= A_k + \frac{2A_k}{\mu}\left[i\left(\frac{A_k}{\mu}\right)^2 + i\left(\frac{B_k}{\mu}\right)^2 + 2m\frac{A_kB_k}{\mu^2}\right] - \left[2i\left(\frac{A_k}{\mu}\right)^2 + 2m\frac{A_kB_k}{\mu^2}\right] \\ &= A_k + \frac{2A_kB_k}{\mu^3}(A_k - B_k)(m-i), \end{aligned}$$

where the first square-bracketed term describes the probability of generating A individuals and the second square-bracketed term describes the probability of A individuals being used as parents. Since $m > i$, the change in A individuals is proportional to $A_k - B_k$. Hence, if there are more A individuals than B individuals in the current population, then it is likely that more A individuals will be in the next generation, and even more in the generation after that, and this cycle is likely to repeat until the population only has A individuals. Likewise, if there are more B individuals than A individuals in the initial population, then it is likely that the population will eventually only have B individuals[3]. Therefore it is expected that the population will rapidly converge toward only having individuals of type A or B due to genetic drift.

This is not the case for mutation. Again assume that there are individuals of type A and B with the same fitness function. The algorithm is still steady state, but now only one child is generated by mutating one parent. Let the probability of a child being generated outside of a hill be d for both types of individuals. The mutation probability is reflected in Table 5.2.

The iterative formula for the number of A individuals in generation $k+1$ for a population of size μ now becomes

$$A_{k+1} = A_k + \frac{A_k}{\mu}\left[d\frac{A_k}{\mu} + d\frac{B_k}{\mu}\right] - \left[d\frac{A_k}{\mu}\right] = A_k.$$

[3]Technically, the system has three equilibria: $A = \mu$, $A = 0$ and $A = B = \mu/2$, with the first two being stable and the last one being unstable.

Parent/Child	A	B	Outside Hill
A	$1-d$	0	d
B	0	$1-d$	d

TABLE 5.2: *The probablity of children being generated by pairs of parents via mutation. The rows represent the parents and the columns the children.*

Thus, for mutation every point is in equilibrium and, unlike for crossover, the population is not expected to converge rapidly toward only having individuals of type A or B. It may be concluded that crossover encourages genetic drift, whereas mutation does not.

5.2.4 Crossover Probability Distribution

It is useful for analytic purposes to approximate the PD of crossover by a continuous distribution, as done for mutation in Section 3.1.2. Deb and Agrawal [46] did exactly this in a paper entitled *Simulated Binary Crossover for Continuous Search Space*, in which they considered a one-dimensional continuous search space. Their analysis is based on a so-called *spread factor* β, which is defined as the absolute ratio of the distance separating children to that separating parents, that is

$$\beta = \left| \frac{c_1 - c_2}{p_1 - p_2} \right|,$$

where c_i and p_j are the positions of the i^{th} child and j^{th} parent, respectively. Taking the extreme strings (individuals with binary values of only 0 or 1) and applying the properties of one-point crossover, they derive a formula for the PD of β as

$$\mathcal{C}(\beta) = \begin{cases} \dfrac{\kappa}{1-\beta} & \text{if } \beta < 1 \\ \dfrac{\kappa}{\beta(\beta-1)} & \text{if } \beta \geq 1, \end{cases}$$

where κ is a constant. The PD of $\mathcal{C}(\beta)$ is shown graphically in Figure 5.6.

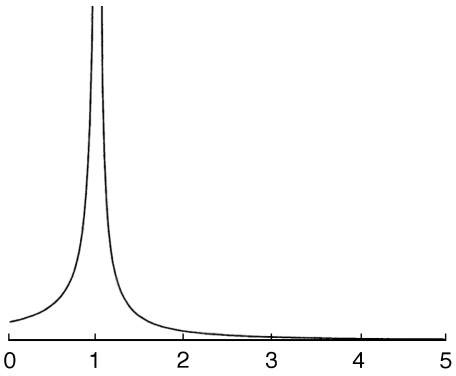


FIGURE 5.6: *The PD of the spread factor β for two extreme binary strings [46].*

Later in the paper the *simulated* binary crossover PD

$$\mathcal{C}(\beta) = \begin{cases} 0.5(m+1)\beta^m & \text{if } \beta < 1 \\ 0.5(m+1)\dfrac{1}{\beta^{m+2}} & \text{if } \beta \geq 1 \end{cases}$$

is proposed, where m is a small integer (*e.g.* $m = 3$). The simulated PD does not have a singularity at $\beta = 1$ and agrees more closely with the PD for non-extreme binary strings, as illustrated in Figure 5.7.

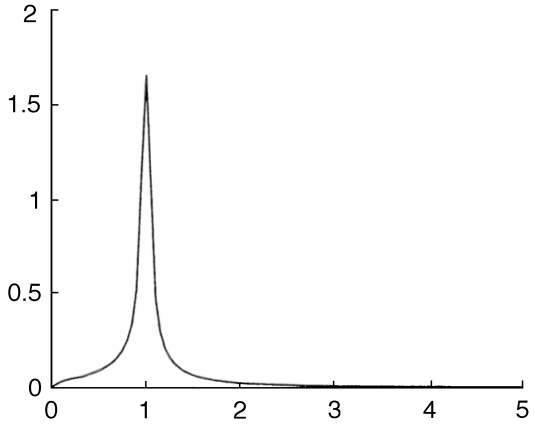


FIGURE 5.7: *The PD of the spread factor β averaged over all pairs of random binary strings of length 15 [46].*

It is crucial to note that these distributions are independent of the distance separating parents. This implies that the step magnitude is proportional to the distance separating parents, agreeing with the distance correlation established in Section 5.2.2.

Since the step magnitude in mutation is independent of the position of individuals, crossover is fundamentally different to mutation. A comparison between continuous crossover and continuous mutation PDs (see Section 3.1.2) may be made by considering two parents in one dimension at positions $x = \pm d$. For the parent situated at position $d > 0$ the continuous PDs for crossover and mutation for $x > 0$ are

$$\begin{aligned} \mathcal{C}(x) &= \begin{cases} \dfrac{-d\kappa}{x-d} & \text{if } 0 < x < d \\ \dfrac{d^2\kappa}{(x-d)x} & \text{if } x \geq d \end{cases} \qquad \text{and} \\ \mathcal{M}(x) &\approx \frac{a}{|x-d|} + \frac{a}{|x+d|}, \end{aligned}$$

respectively, where κ and a are normalised such that the total probability of each distribution is one. For $0 < x < d$, the crossover and mutation distributions are identical except for the scaling, while for $x > 0$, crossover has a lighter tail, as shown in Figure 5.8.

The PDs[4] for continuous crossover, simulated crossover and mutation for parents of various separation distances may be seen in Figures 5.9–5.11. There is a remarkable similarity between the crossover and mutation PDs for parents at positions $x = \pm 0.8$, with the main disparity being that the simulated crossover PD does not have singularities and tends to zero as $x \to 0$. However, the differences become clear for $x = \pm 0.1$ and $x = \pm 4$, as the crossover PDs adapts to the distance between the parents whereas the mutation PD does not. The result is that for parents that are close together (Figure 5.10) mutation has a greater variance than does crossover, while for parents that are far apart (Figure 5.11) crossover exhibits the greater variance.

[4]With κ and a normalised, and $m = 3$ for simulated crossover.

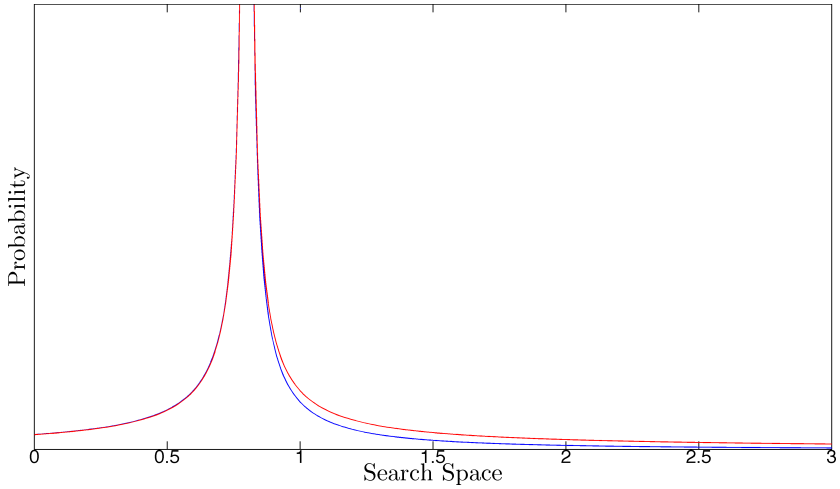


FIGURE 5.8: *Continuous crossover and mutation PDs in blue and red, respectively. The PDs are scaled such that they intercept at the point* $x = 0$.

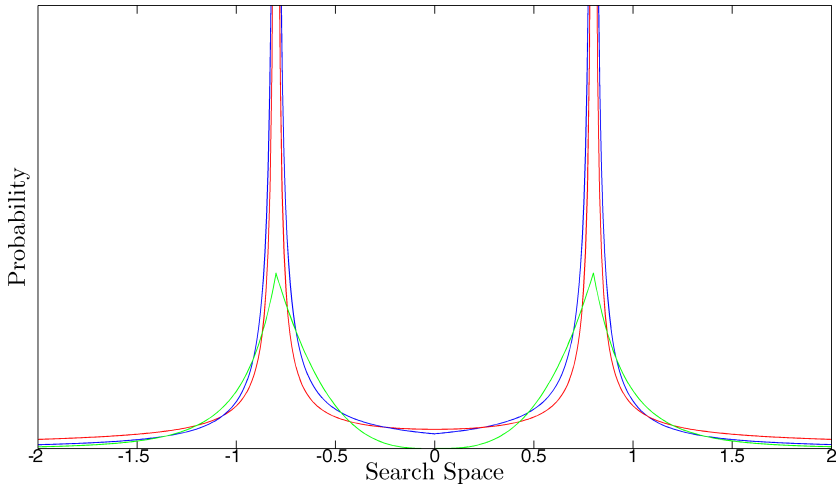


FIGURE 5.9: *Continuous crossover, simulated crossover and mutation PDs in blue, green and red, respectively. The positions of the parents is equal to* $x = \pm 0.8$.

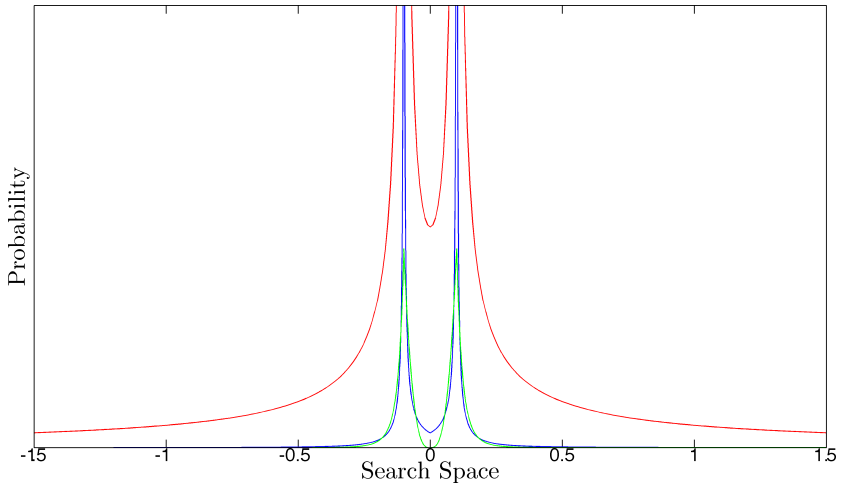


FIGURE 5.10: *Continuous crossover, simulated crossover and mutation PDs in blue, green and red, respectively. The positions of the parents is equal to* $x = \pm 0.1$.

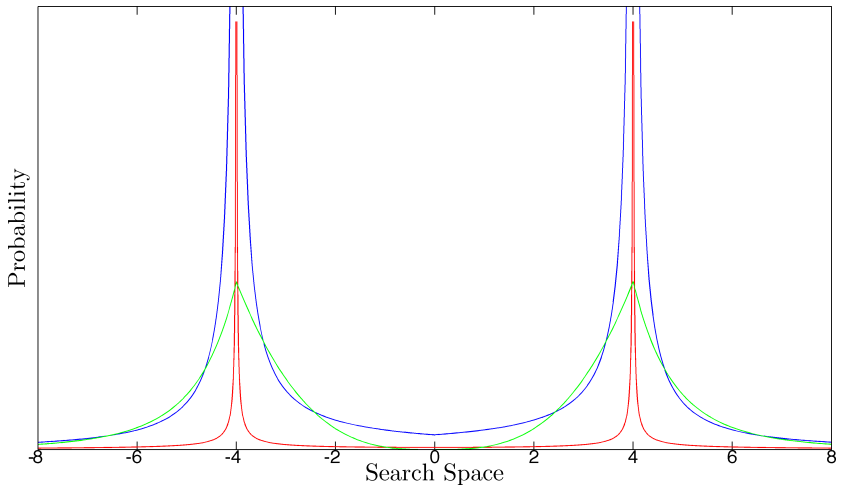


FIGURE 5.11: *Continuous crossover, simulated crossover and mutation PDs in blue, green and red, respectively. The positions of the parents is equal to* $x = \pm 4$.

It may be concluded that crossover and mutation have a similar PD shape in one dimension. The main difference is that for crossover the variance of the PD depends on the distance between the parents, whereas for mutation it does not.

5.2.5 Ellipsoidal Probability Distribution

In the previous subsections the PD of crossover in one dimension was analysed. To complete the analysis of crossover, the step vector in multiple dimensions is now examined. Consider a two-dimensional search space with the x and y dimensions encoded as different bit-strings. The correlation between step magnitude and distance separating parents applies for each dimension separately. Thus, if two individuals have the same x values, then the step vector can only be in the y direction; whereas if two individuals have different x and y values, then the step vector may be in any direction.

The magnitude of the cross product between the step vector and normalised parent difference vector[5] indicates the distance from a child to the straight line extending through its parents[6]. This is a measure of whether the crossover PD is one-dimensionally aligned with the parent difference vector, or if it extends in multiple directions. The cross product was calculated for all pairs of individuals with nine bits, enumerated over all possible combinations and cut positions, and is shown in Figure 5.12.

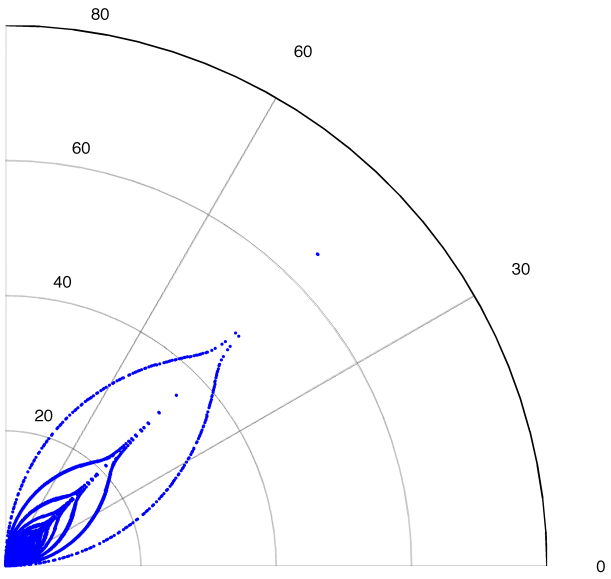


FIGURE 5.12: *Polar plot of the magnitude of the cross product between the step vector and normalised parent difference vector (radius), and the angle of the parent difference vector (angle), for all pairs of individuals with nine bits.*

The figure suggests that the crossover PD is elliptical, with the eccentricity a function of the parent difference vector angle[7]. It reflects the fact that if the angle of the parent difference vector is 0^0 (or 90^0), then the individuals have the same x (or y) values and the step vector

[5]The *parent difference vector* is the vector obtained by subtracting the position of one parent from the other.
[6]The distance for both children is the same according to Section 5.2.1.
[7]Similar observations have been made in the literature (see, for example, [87]). However, the substantiation presented here is novel.

must be in the y (or x) direction and hence, the cross product is zero. On the other hand, if the angle of the parent difference vector is somewhere between 0^0 or 90^0, then the individuals have different x and y values and the step vector may be in any direction, with the result that the cross product may be non-zero.

It follows that the eccentricity of the crossover PD is inversely proportional to the magnitude of the cross product. This is illustrated in Figure 5.13, where the straight line indicates the direction of the difference vector and the ellipse represents the crossover PD.

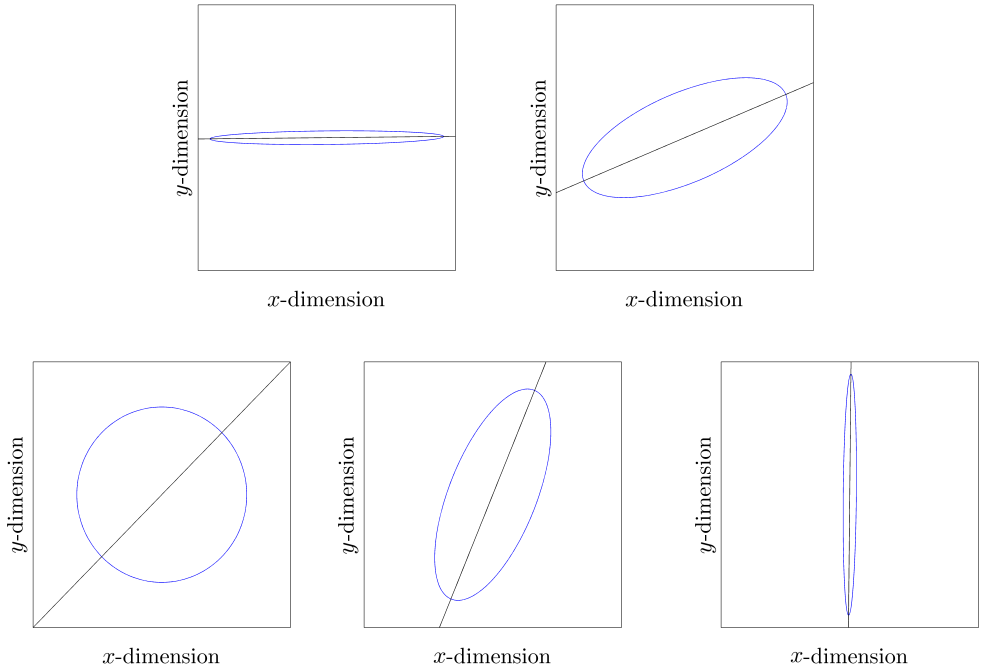


FIGURE 5.13: *Schematic representing the crossover step vector PD for various parent difference vector angles. The straight line indicates the parent difference vector angle and the ellipse the step vector PD.*

A consequence of the PD being elliptical is its invariance under the rotation of 180^0. This ensures that both individuals undergoing crossover have the same step PD (see Section 5.2.1).

5.3 The purpose of crossover

In the previous section the properties of crossover were scrutinised. The attention in this section shifts to the consequences of these properties and how they effect algorithm performance. The analysis is not empirically or mathematically focussed, but instead is largely based on qualitative arguments. Specifically, three qualitative models of crossover are presented and analysed: the Evolutionary Progress Principle and Genetic Repair Hypothesis, the Schema Theorem and Building Block Hypothesis, and the PFL Argument.

5.3.1 The Evolutionary Progress Principle and Genetic Repair Hypothesis

During the 1990s Hans-Georg Beyer proposed the *Genetic Repair Hypothesis* (GRH) in order to show that crossover is an effective operator [15]. It must be noted that he did not focus

on the traditional crossover operator of GAs, but rather on a type of crossover associated with ESs (see Chapter 6). This operator is *mean-centred*, as opposed to *parent-centred*, in that it generates children around the mean position of a number of parents, and not around the position of parents[8]. Since Deb and Agrawal [46] have demonstrated that traditional crossover is parent-centred (see Section 5.2.4), the GRH is not directly applicable to GAs.

The essence behind the GRH is that the mean fitness is less than the fitness of the mean. This may be illustrated by a hypothetical example. Let the fitness function be $f(x, y) = -x^2-(y-t)^2$ and suppose the (infinite) population is distributed in a ring at a distance r away from the origin, with $t \geq (1 + \sqrt{2})r$. Furthermore, assume truncation selection, that is only individuals above a certain level of fitness are selected. The selected individuals are bounded by the angles $\pi/2 - \theta$ and $\pi/2 + \theta$, with $0 < \theta < \pi$, as depicted in Figure 5.14.

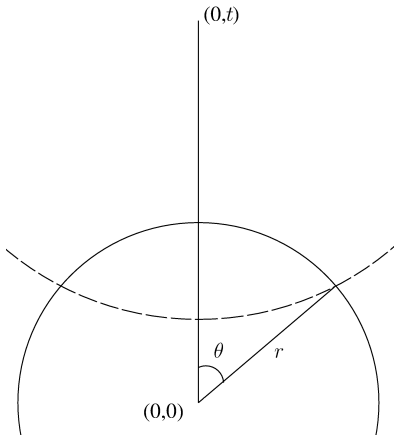


FIGURE 5.14: *A ring of individuals at a distance r away from the origin. The dashed line represents the truncation selection boundary, with only individuals above the boundary surviving selection. The angle θ bounds the selected individuals.*

The mean fitness of the selected individuals is

$$\int_{\pi/2-\theta}^{\pi/2+\theta} -(r\cos\phi)^2 - (r\sin\phi - t)^2 d\phi = -2r^2\theta - 2t^2\theta + 4rt\sin\theta,$$

where the polar coordinate transformation, $x = r\cos\theta$ and $y = r\sin\theta$, has been used. The mean position of the selected individuals is $(0, 2r\sin\theta)$ and, hence, the fitness of the mean is $-4r^2\sin^2\theta$. It may be shown[9] that $-2r^2\theta - 2t^2\theta + 4rt\sin(\theta) \leq -4r^2\sin^2(\theta)$. Therefore, the mean fitness is less than the fitness of the mean for the selected individuals.

This example supports the general hypothesis that the mean fitness is less than the fitness of the mean, at least for a convex fitness function [15]. It suggests that mean-centred crossover, which generates individuals around the mean of a selected population, results in creating individuals of greater fitness and improves the performance of an algorithm.

The other component in Beyer's theory is the *Evolutionary Progress Principle (EPP)*. It states that the evolutionary progress of a search[10] may be decomposed into a positive and a negative part: *evolutionary progress = progress gain – progress loss*. This is motivated by the analytical

[8] There are a number of *Real Coded Genetic Algorithms (RCGAs)* which are either mean or parent-centred [82, 97, 150].

[9] The proof is that $-4r^2\sin^2\theta + 2r^2\theta + 2t^2\theta - 4rt\sin\theta \geq -4r^2\theta + 2r^2\theta + 2t^2\theta - 4rt\theta = 2\theta[(t-r)^2 - 2r^2] \geq 0$.

[10] Evolutionary progress is the rate at which the distance-to-optimum decreases [15].

work of Beyer [19] in which he shows that for ESs the evolutionary progress may be written as $\phi = c_{\mu,\lambda}\sigma - \sigma^2/2$, where $0 \leq c_{\mu,\lambda}$ is the progress coefficient and σ is the mutation strength. Clearly the first term may be equated with progress gain and the second with progress loss. A similar equation may also be derived [18] for ESs with mean-centred crossover, $\phi = c_{\mu/\mu,\lambda}\sigma - \sigma^2/(2\mu)$, where $0 \leq c_{\mu/\mu,\lambda}$. Since the progress loss term for mean-centred crossover is smaller than that of the standard ES by a factor of μ, Beyer concludes that mean-centred crossover decreases the progress loss and thereby increases the evolution progress, making for a more successful algorithm. Note that this conclusion is questionable, as the progress gain term also decreases, $c_{\mu/\mu,\lambda} \leq c_{\mu,\lambda}$.

The EPP and GRH may be summarised as follows: "The progress of a search may be decomposed into two parts, progress gain and progress loss. Crossover decreases progress loss and thereby increases the overall progress." Both of these statements are only partially sound. The first statement, that progress may be decomposed into two parts, is artificial, since the gain and loss parts are correlated. For instance, as the mutation strength increases, both parts increase, while, for mean-centred crossover, both parts decrease. The second statement, that mean-centred crossover decreases progress loss and thereby increases the overall progress, is also questionable. This a consequence of the invalidity of the first statement (mean-centred crossover also decreases progress gain) as well as the fact that GR can only be shown to be effective for convex fitness functions.

Even though the EPP and GRH may be mathematically formulated, they are not a mathematical proof, but rather a QM. They do shed some light on evolutionary progress, but do not formally prove any results about algorithmic performance.

An extension of these arguments is proposed for GAs, although Beyer admits that "there is no direct and totally satisfactory way" to do this [15]. However, "some indirect evidences exist," such as: the relation between the mutation rate in GAs and the mutation strength in ESs, the mathematical results of the `OneMax`-function and the empirical results of multi-mixing GAs [15]. Thus, the EPP and GRH may be tentatively proposed as explanations for GAs.

5.3.2 Schema Theorem and Building Block Hypothesis

The *Schema Theorem* was proposed by Holland [83] in the early days of GAs and was for a long while the dominant explanation of crossover. It considers *schemata*, which are the set of strings in the space $\{0, 1, *\}^k$, where k is the length of the bit-string and the wildcard symbol $*$ may represent either a 0 or 1. For example, the schema $1*0*$ refers to the set of four strings $\{1000, 1001, 1100, 1101\}$.

The Schema Theorem estimates the expected number of instances of schema H at time t, denoted by $m(H, t)$. In the absence of mutation or crossover, fitness proportional selection causes the expected number of instances of a schema to be

$$E[m(H, t+1)] = \frac{\sum_{x \in H} f(x)}{|H|\bar{f}(t)} m(H, t),$$

where $|H|$ is the size of the schema set H, f is the fitness function and $\bar{f}(t)$ is the average fitness of the population at time t.

The probability of a schema surviving mutation is $(1 - p_m)^{o(H)}$, where $o(H)$ is the order of the schema (the number of non-wildcards in the schema) and p_m is the mutation rate. This is due to each bit in the schema having a probability of $1 - p_m$ of not being flipped. For single-point crossover, the probability of a schema surviving is bounded from below by $1 - p_c \cdot d_H/(k-1)$,

where p_c is the crossover probability and d_H is the defining length of the schema[11] and k is the bit-string length. The reason for the bound, as opposed to equality, is that it is possible for crossover to cut a schema without it being altered, for example, if two members of the same schema undergo crossover.

Combining the effects of selection, mutation and crossover, the Schema Theorem states that

$$E[m(H,t+1)] \geq \frac{\sum_{x\in H} f(x)}{|H|\bar{f}(t)} m(H,t) \times (1-p_m)^{o(H)} \times \left(1 - p_c \cdot \frac{d_H}{k-1}\right).$$

The inequality implies that short (d_H is small) and low-order ($o(H)$ is small) schemata of above average fitness are likely to survive, an idea that was formalised in the *Building Block Hypothesis* (BBH).

A *Building Block* (BB) is a short, low-order schema of above average fitness. According to the Schema Theorem, the number of instances of a BB is expected to increase over each generation. Since the population is expected to converge toward the maximum solution, it is claimed that the maximum solution must be composed of these BBs. This idea was expressed by Goldberg as follows:

> "Just as a child creates a magnificent fortress through the arrangement of simple blocks of wood, so does a genetic algorithm seek near optimal performance through the juxtaposition of short, low-order schemas or BBs" [68] (according to [157]).

In this sense crossover focusses on the level of the schema, not the individual. From this viewpoint, each individual may be thought of as a collection of schemata. When two parents undergo crossover, all of their schemata are effectively operated on in parallel, a process known as *implicit parallelism* [157]. If the children's schemata are fit, then the children will survive selection and the schemata will be passed onto the next generation[12]. Since implicit parallelism with BBs is unique to crossover, it follows that crossover has the unique ability to progress the search.

There are a number of arguments against the BBH. The first is that the *observed* fitness of a schema is different to its *actual* fitness. The actual fitness of a schema is equal to the average fitness of its elements, $f(H) = \sum_{x\in H} f(x)/|H|$. The observed fitness, on the other hand, refers to the average fitness of the elements of a schema present in the current population. This value may vary due to the variety of the fitness values of elements in the schema and is exacerbated by the prevalence of certain schemata in the population. For example, consider the fitness function $f(x) = -|x-8|$ and the schema $***1$, where all individuals in the population are of the schema $11**$. Even though $***1$ has an actual fitness of -4, combined with the schema $11**$, it forms the schema $11*1$ which has an actual fitness of -6. Therefore, in a population where all individuals are of the schema $11**$, the schema $***1$ has the observed fitness[13] of -6, even though its actual fitness is -4. Phrased more generally, if certain schemata are ubiquitous in the population, then the actual fitness becomes a poor indicator of observed fitness.

Other criticisms of the BBH include its failure to estimate the population schema proportions in problems with noise and other effects [60] (although this point is debatable [135]), the fact that the Schema Theorem itself shows that BBs may be detrimental (for instance, in the case

[11] The *defining length* of a schema is the position of the last non-wildcard bit minus that of the first non-wildcard bit. For example, the defining length of $1*0*$ is $3-1=2$.

[12] This agrees with the modern biological paradigm of genes, most prominently popularised by Richard Dawkin's book *The selfish gene* [43].

[13] The observed fitness may differ from -6 depending on exactly which instances of the schema $11*1$ are in the population.

of deception [72]), and finally that it does not consider the constructive effects of crossover or mutation.

The so-called *construction* and *disruption theories* have been developed to address this final shortcoming. These respective theories involve calculating the probability of a schema being constructed (created) or disrupted (destroyed). Although they are connected — in order to construct new schema an old schema must be disrupted — they have a slightly different focus and require separate analysis [162]. The construction and disruption probabilities for crossover and mutation may be seen in Figure 5.15, reproduced from [162].

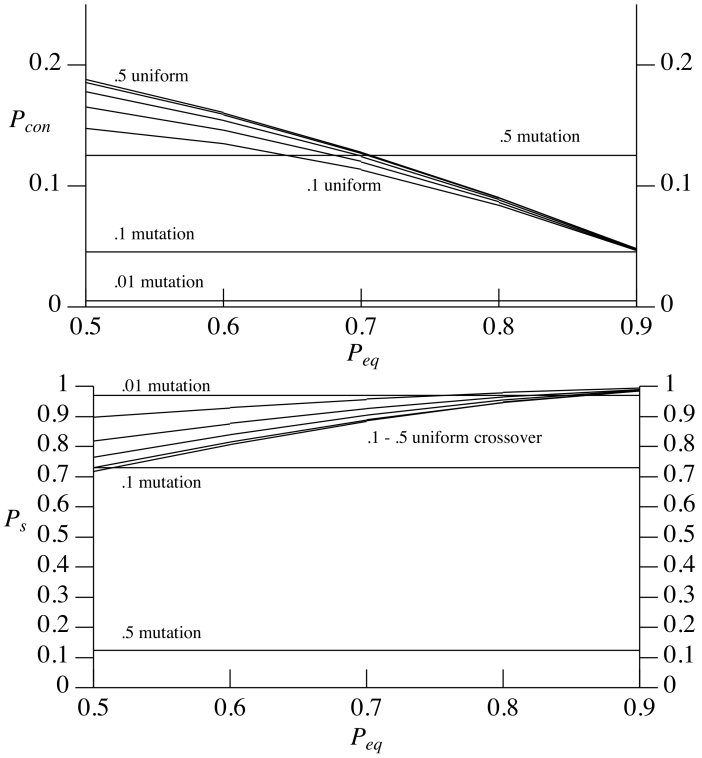


FIGURE 5.15: *The probability of construction P_{con} and probability of survival P_s (the opposite of the probability of disruption) versus the level of convergence P_{eq} (for definitions of P_{con}, P_s and P_{eq}, see [162]). The labels* .1 uniform *to* .5 uniform *refer to uniform crossover with the probability of swapping bits being 0.1 to 0.5, respectively. Likewise,* .01 mutation *to* .5 mutation *refer to the probability of flipping bits being 0.01 to 0.5, respectively.*

The main conclusion that may be drawn is that crossover is highly dependent on the convergence of the population, whereas mutation is not. Initially, when the population has not converged, the construction (and disruption) rates of crossover are higher than that of mutation. This is because initially the population is diverse and individuals undergoing crossover typically differ on many bits, with the result that crossover frequently flips more bits than if the individuals underwent mutation separately[14]. As the population converges, the probability of individuals differing in many bits decreases and, accordingly, the frequency of bits flipped during crossover decreases until eventually, when the population has converged, crossover is incapable of any construction (or disruption). This process may occur for individual bit positions, a phenomenon known as *losing bits*. Bits are lost when selection leads to all individuals in the population

[14]This argument relies on crossover being seen as a type of mutation, as has been argued by Fogel [58].

having either a 0 or 1 in a certain bit position, in which case crossover is unable to construct individuals with either bit value. On the other hand, mutation has the same construction (or disruption) rate, no matter how converged the population is, and is not affected by losing bits.

The Schema Theorem and BBH present an intuitively appealing explanation of GAs. It may be summarised as follows: "A maximum solution is comprised of BBs, which are short, low-order schemata of high average fitness. Throughout a search, crossover causes these BBs to increase in frequency until the population converges to a maximum solution." The Schema Theorem and BBH, just like the EPP and GRH, is actually a QM. They too are based on two principles: firstly, that a maximum solution may be decomposed into BBs and secondly, that crossover causes BBs to increase in frequency throughout a search. Both of these principles are generally, but not always, sound. The first principle fails in the case of deception, when the maximum solution is not composed of BBs of high fitness, and the second principle often fails due to the stochasticity of selection and variation in fitness within a schema.

Moreover, even if the above principles do account for the workings of crossover, they are not unique to crossover. As remarked by Beyer [15],

> "They hold for any iterative procedure that successively approaches an optimum, if BBs are interpreted as certain decompositions of the optimum solution. The exponential growth of schemata can even be observed in gradient strategies under such conditions."

This view agrees with the sentiments of Fogel [58], who claims that "rather than being fundamentally different from random mutation, as claimed, crossover [is] merely a subset of all random mutations." Hence, there is no explanation of algorithm performance unique to crossover. This conclusion should not necessarily be viewed as a failure of the Schema Theorem and BBH, but rather as further substantiation that crossover is simply a specific type of mutation.

5.3.3 The PFL Argument

In section 5.2 it was established that crossover

1. has opposite step vectors,
2. is parent-centred with its variance proportional to the distance between parents,
3. promotes genetic drift, and
4. has an ellipsoidal PD.

The effects of these properties may be deduced according to the framework developed in Chapter 4. Specifically, as discussed in Section 4.5, the effectiveness of crossover and its properties may be judged on three criteria: the agreement of the PD with the IPD, population diversity and computational speed.

The effect of crossover on computational speed may be determined without considering the consequences of the above-mentioned properties. As crossover increases the number of computation steps required to produce an individual, it slows down the computational speed of the algorithm. However, this increase is typically[15] insignificant as the speed is usually limited by the number

[15]It may not always be the case that the number of fitness function evaluations is the main limiting factor of the computational speed, in which case the use of crossover could significantly slow the algorithm down.

of fitness function evaluations, not the number of computation steps required to generate an individual. Thus, the effect of crossover on computational speed is negligible. The focus in the rest of this section will fall on how each crossover property affects the agreement of the PD with the IPD and population diversity.

Opposite step vectors

The first property to consider is that for two individuals undergoing crossover, the step vector of the one individual is the opposite of the other (see Section 5.2.1). A 180^0 rotation of the step vector is the optimum angle for increasing the distance between the children; therefore, this property maximises the distance between children without affecting the PD. Since a large distance between children leads to a more diverse population, this property has a beneficial effect on algorithmic performance.

Parent-centred and variance proportional to the distance between parents

The second property (see Sections 5.2.2 and 5.2.4) motivates a comparison between mutation and crossover. Mutation and crossover are both parent-centred, although the variance for mutation is fixed, whereas for crossover it is proportional to the distance between parents. This was observed in [47] and García-Martínez *et al.* [61] went on to "conclude that [crossover] may be seen as self-adaptive ... mutation," where "[crossover] calculates implicitly the [variance] using information about the distribution of the individuals in the population."[16] The result is that the variance for crossover is smaller than that for mutation in high-density regions, and larger in low-density regions, as illustrated in Figure 5.16.

The dependence of variance on density agrees nicely with schemata-based construction (and disruption) theory (see Section 5.3.2), which states that the constructive (or disruptive) power of crossover is inversely proportional to the degree of population convergence, whereas the power for mutation is independent of population convergence.

A smaller variance is desirable in high-density regions for two reasons. Firstly, this decreases the overlap of neighbouring individuals' PDs, which increases diversity (see Section 4.6.1). Secondly, due to selection there is a correlation between the density of individuals and their fitness, specifically in higher-density regions individuals are typically fitter. In the PFL the expected fitness around fitter individuals decreases faster, as there is a greater difference between the fitness of the individual and the average fitness of the search space (see Section 4.2.3 for an illustration of this phenomenon). Thus, for fitter individuals, points of relatively[17] high expected fitness are closer. Since individuals in higher-density regions are typically fitter, points of relatively high expected fitness are closer. Therefore, in order to generate points of high expected fitness in high-density regions the variance should be smaller.

On the other hand, it is also desirable to have a large variance in low-density regions. Again, there are two reasons. Firstly, it increases ergodicity (the ability to reach any point in the search space [53]) which leads to greater diversity. Secondly, the inverse of the density-fitness correlation argument may be used to motivate generating points of relatively high expected fitness far away from individuals in low-density regions. Note that it is occasionally undesirable to have a large

[16] The bracketed *crossover* is in place of the phrase *PCCOs* (Parent-Centric Crossover Operators) and *variance* replaces *standard deviations*.

[17] Note that the term *relatively* is essential. All of the points around a fitter individual are of higher expected fitness; however, *relative* to the fitness of the individual this is not the case. Points close to a fit individual are relatively more fit, whereas points far from a fit individual are relatively less fit, than for an unfit individual.

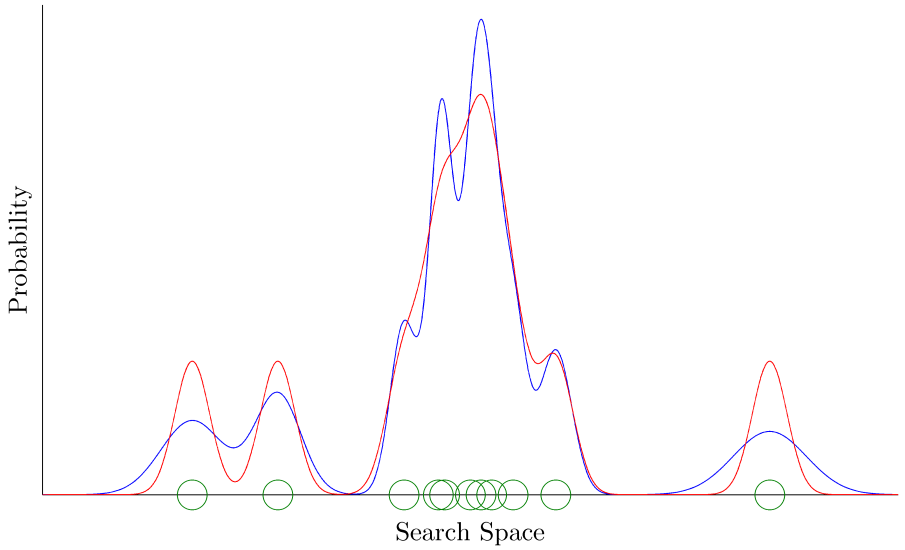


FIGURE 5.16: *The mutation and crossover PDs of a population shown in blue and red, respectively. The green circles represent the positions of individuals in the population. For mutation every individual's PD is a Gaussian mutation of fixed variance, whereas for crossover the PD of each individual is also a Gaussian, but its variance is proportional to its average distance to all other individuals in the population.*

variance in low-density regions. For example, a large mutation may lead to an individual being generated in a region that has not yet been visited by the population. This region may be of high fitness (perhaps containing the global maximum) and the generated individual may have a very high fitness value. In this case it is desirable to create more individuals in this region, close to the previously generated individual, and hence a small variance is required.

Since a smaller variance in high-density regions and a larger variance in low-density regions both increase diversity and generally improve the agreement of the PD with the IPD, it may be concluded that the ability to adapt variance according to density is generally beneficial.

Promotion of genetic drift

Genetic drift (see Section 5.2.3) only occurs in certain situations, namely when there are *species*, with inter-species crossover frequently producing *lethals*[18]. To decode this statement, a species is an isolated set of individuals in the population occupying a region, typically around a local maximum[19]. There may be several species in one population. Crossover between fit parents of different species often lead to unfit children, known as lethals [151] — although each parent may be fit, the children may be unfit if they fall in the valley between the species' local maxima [14].

If there is genetic drift, as demonstrated in Section 5.2.3, a single species will increasingly dominate until the whole population eventually converges to it (even though there may exist multiple maxima of equal or greater fitness). Clearly a population with only one species is less

[18]The notion of "inter-species crossover frequently producing lethals" neatly agrees with the biological definition of a species as "a set of individuals which may breed together to produce viable offspring" [14].

[19]Such a region is known as a *niche*.

diverse. Since this does not occur for mutation, it may be concluded that in certain situations crossover promotes genetic drift and thereby decreases diversity.

Genetic drift may also be observed from the perspective of schemata, where schemata are equivalent to species. Crossover between fit parents belonging to different schemata may be lead to unfit children. For example, consider the fitness function $f(x) = -|x-8|$ and schemata $1***$ and $0***$. The individuals 1000 and 0111 are fit, having the respective fitness values of 0 and -1, yet if they undergo crossover they may produce the children 1111 and 0000, with respective fitness values of -7 and -8. As is the case with species, one schema will increasingly dominate until the whole population eventually converges to it (even though there may exist multiple schemata of equal or greater fitness), thereby decreasing diversity.

Ellipsoidal Probability Distribution

An ellipsoidal PD (see Section 5.2.5) allows the variance of the step vector to be different along the different axes of the search space. If the fitness function may be decomposed into separate functions along the axes of the search space, then this enables each decomposed function to be searched with its own appropriate variance. Therefore, the PD will be in greater agreement with the IPD, which is beneficial. However, if the axes of the search space do not align with the decomposition of the fitness function (for example, if a hill is parallel with the line $y = x$), then the ellipsoidal PD is of no benefit.

Summary of the above four properties

Four properties of crossover were established in Section 5.2, two of which (variance proportional to the distance between the parents and the promotion of genetic drift) were further substantiated via schemata analysis. The consequences of the four properties may now be viewed in combination to determine the overall effect of crossover. Diversity is generally increased by creating children that are further apart and by the variance being proportional to the distance between parents, but it may decrease in the case of genetic drift. The agreement of the PD with the IPD may also increase due to the variance being proportional to the distance between parents and the ellipsoidal PD. Therefore it is concluded that, although it occasionally has a neutral or detrimental effect, crossover may be generally beneficial.

5.4 Comparison of Qualitative Models

The above summary of the properties of crossover is intentionally tentative in respect of the *No Free Lunch (NFL) theorem* [180] (discussed in [163, 181]). The theorem "state[s] that any two optimisation algorithms are equivalent when their performance is averaged across all possible problems" [179]. Therefore a phrase such as, "crossover is generally beneficial" is technically invalid, since every operator is as good as any other operator in general. Fortunately, the problems which are typically found in the real world form a subset of "all possible problems" and hence the NFL theorem does not quite apply to this subset [62]. Even so, for any particular problem (or *problem class*[20]) the theorem implies that no algorithm may be said to be better *a priori*, but must be demonstrated to be so mathematically, empirically or by a QM.

[20] A *problem class* is a set of problems which share similar characteristics. Examples of problem classes include: unimodal, bimodal or multi-modal fitness functions.

The EPP and GRH, the Schema Theorem and BBH, as well as the novel PFL Argument are all QMs of crossover. Each of them is based on two principles, shown in Table 5.3. Since the PFL Argument pertains to all metaheuristics, its implementation to crossover depends on the properties of crossover discussed in the previous section.

Qualitative Model	Principle 1	Principle 2
EPP and GR	Evolutionary progress = progress gain – progress loss	Crossover causes a decrease in progress loss
Schema Theorem and BBH	A maximum solution may be decomposed into BBs	Crossover causes BBs to increase in frequency
PFL Argument	Individuals in the population have above average fitness	The further away a point is from an individual, the larger the range of its possible fitness values

TABLE 5.3: *The principles of Qualitative Models for crossover.*

The notions of a *valid* and *sound* argument are now introduced in order to analyse the three QMs mentioned above. An argument is *valid* if it takes the form that if its premises are true, then the conclusion is true. On the other hand, an argument is *sound* if it is both valid and all of its premises are actually true [86]. An example of a valid argument is: "all fish have gills." Now if this argument is combined with the premise that "goldfish have gills" and it is concluded that "goldfish are fish," then the argument is clearly sound. However, if the argument has the false premise that "giraffes have gills" and accordingly concludes that "giraffes are fish," then the conclusion is obviously false and the argument is not sound. Thus, even if an argument is valid, it is not sound when combined with a false premise.

In the case of the three QMs mentioned above the principles are equivalent to premises. If, for a particular problem, the principles hold true and the QM is valid, then the QM is sound and should be able to predict (in foresight) or explain (in hindsight) an algorithm's performance. On the other hand, if the principles do not hold true, or the QM is invalid, then the QM cannot be applied as the argument will be unsound. There are two techniques to determine whether a QM is sound: to assess the truth of its conclusion or to examine the validity of its argument. For the QMs of crossover this may be achieved by an empirical, mathematical or qualitative analysis.

Empirically, there are multiple papers (for example, [58, 110]) that analyse the performance of crossover, each focussing on different problems and reporting different results, some of which are in favour of crossover and others not. The process of empirical testing has in practice had its own problems [84, 92], making it difficult to arrive at any grand conclusions. Overall, the empirical evidence seems to align with the NFL theorem: that crossover is not generally beneficial, although there are certain problems (and problem classes) for which it does improve performance.

Mathematically, an exact model of GAs[21] was solved by Vose and Liepins [172] (according to [157]). The solution suggested that there were several, sometimes conflicting, mechanisms simultaneously manifesting themselves in GAs, of which both crossover and mutation were simultaneously a part — a direct contradiction of the EPP. They concluded that operators play an insignificant role in GAs and that crossover and mutation are not much different.

This view was supported by Fogel [58] who denied the claim that "sophisticated genetic operators

[21] The GA considered in [172] had an infinite population.

are required to ensure successful adaptation." He argued that since crossover just flips bits, it is effectively a form of mutation and therefore could not have a significant advantage over it. In fact, from the perspective of EAs, there is little reason to believe that crossover should work at all. The grand QM of EAs is that of Universal Darwinism (see Chapter 1), which states that the fitness of individuals is expected to increase if there is inheritance, selection and variation. Since crossover does not always maintain inheritance[22], Universal Darwinism fails and there is no reason to believe that an algorithm with crossover leads to an increase in fitness.

The lack of empirical and mathematical evidence, as well as the fact that crossover violates the QM of Universal Darwinism, suggests that crossover is generally not of any benefit (or only of slight benefit). Therefore a QM concluding that crossover is generally beneficial must be generally unsound. It remains unclear whether this is due to the QM's principles generally not holding true or its argument being invalid, or a combination of the two. Further analysis is required in order to distinguish which is the case for each QM.

The EPP and GRH

The EPP and GRH QM proposes that crossover is generally beneficial and is therefore generally unsound. In fact it is both invalid and its principles do not generally hold true. Its invalidity stems, firstly, from Vose and Liepins's mathematical results [172] which show the EPP to be invalid and, secondly, from the GRH failing to recognise that crossover also decreases progress gain. On top of this, the GRH only holds true for convex fitness functions. Thus, although the advice of Beyer [15] to "identify the gain/loss parts and try to increase/decrease them" may be useful, his QM is very weak.

The Schema Theorem and BBH

The Schema Theorem and BBH QM also predicts that crossover is generally beneficial and is accordingly only sometimes sound. Although both of its principles are valid, the one generally holds true, while the other only holds true some of the time.

The generally sound principle is that crossover causes BBs to increase in frequency. This is because it holds true for all problems and is valid, with the caveat of observed fitness, stochastic variation and the question of whether the increase in BB frequency is caused by crossover, not mutation. It is even sound for deceptive problems where crossover is detrimental. However, the principle that a maximum solution may be decomposed into BBs does not hold true for some problems (*e.g.* deceptive problems). In fact, it is probably the case that this principle only holds true when crossover is beneficial and as a result only holds true roughly half of the time. Therefore, even though the principle may be valid, it is only sometimes sound. Consequentially the entire QM, although valid, is only sound some of the time.

If a QM consisted of just the principle that crossover causes BBs to increase in frequency, then it would be generally sound. Since it would not predict that crossover is generally beneficial, it would be consistent with the prevailing empirical and mathematical findings. This would leave the maximum solution principle to be used as a technique to identify problems for which crossover is expected to be beneficial.

[22] Crossover occasionally flips many more bits than mutation usually would, creating children that have no link their parents.

The PFL Argument

The PFL Argument is the only QM not to predict that crossover is generally beneficial and therefore may be generally sound. Since the PFL Argument is phrased in general terms, its principles should always hold true, and therefore its soundness solely depends on its validity. The fact that the Schema Theorem and BBH agree with the PFL Argument (that crossover is parent-centred with its variance proportional to the distance between parents and promotes genetic drift) reinforces the validity of both QMs. However, this is not enough to show that the PFL is sound.

The ultimate test of validity is mathematical or empirical. A mathematical proof would guarantee the validity of the PFL Argument, although it is unlikely that such a proof exists for EAs (see Chapter 1). The empirical approach would involve systematically considering problems with certain properties and testing whether the PFL Argument correctly predicts the behaviour of the algorithm based on those properties. An example of such a test would be a multimodal fitness landscape with one global maximum that is of slightly higher fitness than the rest of the local maxima. The PFL Argument predicts that in this kind of problem the population is likely to be divided into many species and that genetic drift, which is encouraged by crossover, would cause the entire population to converge to the local maximum of one species, even though its local maximum might not be the global maximum. Hence, crossover would lead to a less diverse search with worse performance than an algorithm that does not use crossover. Alternatively, for a unimodal fitness landscape genetic drift is not problematic and, due to all of the beneficial properties of crossover, it should lead to better performance. Such empirical simulations would demonstrate whether the PFL Argument is valid in those cases, and thereby it may be induced whether it is generally valid or not. Although this would be very valuable, such an analysis is beyond the scope of this thesis, but may be pursued in further work.

It may be the case that the PFL Argument is invalid if it does not accurately predict the behaviour of crossover for certain problems. This could simply be because the properties upon which the PFL Argument depends do not capture all of the behaviour of crossover[23]. The uncaptured behaviour may be attributed to properties that have not yet been recognised. If these unrecognised properties are identified and incorporated into the PFL Argument, then this would improve the validity of the PFL Argument. Hence, the PFL Argument may be improved over time as more properties of crossover are identified.

5.5 Chapter summary

This chapter has introduced GAs and accordingly focussed on the operator unique to GAs, crossover. The different types of crossover were described after which the properties of crossover were analysed. Four properties were identified and examined: opposite step vectors, parent-centred and variance proportional to the distance between parents,the promotion of genetic drift and the ellipsoidal PD. This was followed by an investigation of the purpose of crossover via three QMs: the EPP and GRH, the Schema Theorem and BBH, and the PFL Argument.

Out of the three QMs of crossover, the EPP and GRH is clearly the weakest and may be rejected. The Schema Theorem and BBH comprise the original QM and the fact that it is still popular is testament to its power. However, it suffers from the outdated notion that crossover is beneficial. Fortunately it may be updated by separating its two principles so that the one forms a generally sound QM, while the other becomes a test for when the use of crossover may be beneficial. This

[23] This argument is an application of the Duhem-Quine thesis.

is exactly what a good QM should be: able to identify properties of algorithms and thereby predict their behaviour with respect to particular problems, recognising when certain operators might be beneficial. The PFL Argument is a pure version of such a QM. It is not so much a QM of crossover, but a platform for properties of crossover to be scrutenised. In this chapter four properties were identified and analysed, pointing to some of the advantages and disadvantages of crossover. These four properties may not form a complete description of crossover and as new properties are recognised they may be incorporated into the PFL Argument, improving it further.

It is unclear whether the Schema Theorem and BBH QM or the PFL Argument is superior. Fortunately this question is of no relevance. What is important is that both QMs may be used to explain and predict the performance of crossover. For some problems it may be the case that it is more easily analysed via the Schema Theorem and BBH QM, whereas for other problems the PFL Argument may be more amenable. Instead of being viewed as competitors, the QMs should be thought of as complementary, which together may provide a better understanding the nature of crossover.

CHAPTER 6

Evolution Strategies and Evolutionary Programming

Contents

Evolution Strategies (ESs) and *Evolutionary Programming* (EP) are two peas in a pod, although the pod did not know this for a very long time. Both of these EAs were developed during the 1960s but on different sides of the Atlantic Ocean. ESs were largely the product of Ingo Rechenberg, Hans-Paul Schwefel and Peter Bienert who did much of their seminal work at the Technical University of Berlin in Germany [6], while Lawrence J Fogel devised EP at the National Science Foundation in the USA [59]. Although ESs and EP are very similar, their respective research communities only established formal contact during the early 1990s, nearly a decade after the ES and GA communities began communicating [8].

The similarities between ESs and EP are so great that they have now practically merged into one field. They both operate on real values, use a Gaussian distribution to mutate individuals in the population (see Section 3.1.1) and, most importantly, employ self-adaptive parameters. However, they do have their differences. ESs use (μ, λ) or $(\mu + \lambda)$ truncation selection, while EP uses $(\mu + \lambda)$ tournament selection (see Section 3.2), and ESs have a recombination operator whereas EP has none. In the following sections, each algorithm is examined in greater detail, after which the QM of *effective fitness* is presented to explain why they work.

6.1 Evolution Strategies

The key features of ESs are their self-adaptive mutation operators and reproduction operators. Mutation is investigated in the following subsection, after which recombination is considered.

6.1.1 Mutation

Mutation in ESs is based on a Gaussian distribution, and if the search space is one-dimensional, then a symmetrical one-dimensional Gaussian distribution is used. However, if there are multiple dimensions then a symmetric distribution may not be desirable — it may be the case that the mutation PD should be ellipsoidal. For example, if there are narrow hills in the fitness landscape, then the IPD would be elliptically aligned along these hills [6, p.69].

Crossover may cause the PD to be elliptical by employing different variances along the different axes of the search space (see Section 5.2.5). ESs achieve a slightly more impressive feat[1] by using a *covariance matrix* to generate step vectors according to the ellipsoidal PD

$$p(\Delta s) = G(\vec{0}, \mathbf{C}) \equiv \frac{\exp\left(-\frac{1}{2}\Delta s^T \mathbf{C}^{-1} \Delta s\right)}{\sqrt{(2\pi)^n \cdot \det \mathbf{C}}}, \tag{6.1}$$

where G denotes the generalised Gaussian distribution with expectation $\vec{0}$ and covariance matrix $\mathbf{C}^{-1}$, Δs is the mutation step vector and Δs^T is its transpose [6]. This distribution is, as its name suggests, a generalisation of the Gaussian distribution. In fact, if $\mathbf{C}$ is a diagonal matrix with entries $\sigma_1, \sigma_2, \ldots, \sigma_n$ along the diagonal, then the PD is equivalent to a multidimensional Gaussian distribution with variance σ_i^2 along the i^{th} coordinate axis. If these variances are not equal, then the PD will not be spherical but ellipsoidal, and if $\mathbf{C}^{-1}$ has non-zero non-diagonal elements, then the ellipsoids will not be aligned with the search space axes. A graphical representation of ellipsoidal PDs is reproduced from [6] in Figure 6.1.

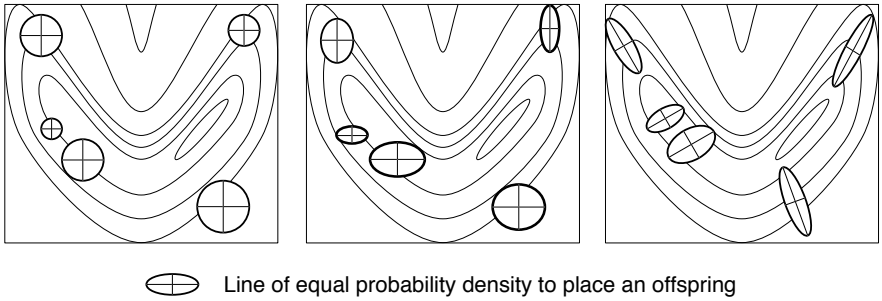


FIGURE 6.1: *Illustration of ellipsoidal mutation PDs. In each graphic five individuals and their ellipsoidal PDs are shown on a fitness landscape, which is represented by contours of equal value. The covariance matrix for each individual in the left most graphic is diagonal with equal variances, in the middle graphic is diagonal with different variances and in the right most graphic is non-diagonal [6].*

Generally speaking, the covariance matrix $\mathbf{C}^{-1}$ is an $n \times n$ symmetric matrix which describes the relationship between the coordinates in the PD. Practically it is advisable not to encode the elements in the covariance matrix directly, but instead to use the angles of the ellipsoid α_{ij} and the magnitude of the variances σ_i^2. The relation between the entries in the covariance matrix c_{ij} and the angles α_{ij} and variances σ_k is given by

$$\tan(2\alpha_{ij}) = \frac{2c_{ij}}{\sigma_i^2 - \sigma_j^2},$$

[1] The ES elliptical PD is more general than that of crossover as it need not be aligned with the search space axes.

where[2] $i \neq j$. Practically this transfomration is achieved using rotation matrices. Each rotation matrix $\mathbf{R}$ corresponds to an angle α_{ij}, therefore $\mathbf{R} = \mathbf{R}(\alpha_{ij})$. Its entries at position (k, ℓ) are given by

$$\mathbf{R}(\alpha_{ij})_{k\ell} = \begin{cases} \cos(\alpha_{ij}) & \text{if } k\ell = ii \text{ or if } k\ell = jj \\ -\sin(\alpha_{ij}) & \text{if } k\ell = ij \\ \sin(\alpha_{ij}) & \text{if } k\ell = ji \\ \delta_{ij} & \text{otherwise,} \end{cases}$$

where the Kronecker delta function δ_{ij} takes the value of 1 if $i = j$, or 0 otherwise. A step vector may be generated according to the PD in (6.1) with angles α_{ij} and variances σ_k via a two step process. Firstly, an uncorrelated step vector should be generated whose k^{th} element is generated by a one-dimensional Gaussian of variance σ_k^2. Secondly, this vector should be multiplied by each of the rotation matrices (in any order) to obtain the correlated step vector. Using this process a correlated step vector may be generated without directly using the correlation matrix.

The point of this roundabout method is to render the correlation matrix easily adaptable. As was discussed in Sections 2.4.2 and 2.4.3, it is beneficial for the PD to be self-adaptive, that is, to be able to change in response to how the algorithm is performing during the search. In the case of ESs, self-adaption involves adapting the entries in the covariance matrix. The problem with adapting the covariance matrix directly is that it is difficult to guarantee that the coordinate system remains positive definite[3]. This problem may be avoided by using angles and variances.

In ESs the angles and variances are adapted like the positions of individuals — via mutation. Elementary algorithms, such as Fixed step size random search or the Matyas method (see Section 2.4), only have one variable associated with each individual, namely its position. In ESs each individual has two additional sets of variables associated with it: angles and variances. In other words, each individual has its own PD. Each individual is assigned a triple vector $(\vec{s}, \vec{\alpha}, \vec{\sigma})$, representing its position, angles and variances. During each iteration, all of the variables of every individual in the population are adapted via mutation. The equations for mutating an individual are

$$\begin{aligned} \sigma_i' &= \sigma_i \cdot \exp(\tau' \cdot G(0,1) + \tau \cdot G_i(0,1)), \\ \alpha_j' &= \alpha_j + \beta \cdot G_j(0,1) \qquad \text{and} \\ \vec{s}\,' &= \vec{s} + \vec{G}(\vec{0}, \mathbf{C}) \end{aligned}$$

for all variances $i \in \{1, \ldots, n\}$ and angles $j \in \{1, \ldots, n \cdot (n-1)/2\}$. The parameters τ, τ' and β control the total variance, indexed variance and individual angle mutations. They are usually set to the values $\tau \propto (2\sqrt{N})^{-1/2}$, $\tau' \propto (2N)^{-1/2}$ and $\beta = 5\,\text{rad} \approx 0.0873°$, where N is the number of iterations [6].

The notation $G(0,1)$ is used to denote the realisation of a Gaussian distributed one-dimensional random variable having expectation zero and variance one, while $G_i(0,1)$ indicates that the random variable is sampled anew for each possible value of the counter i. Note that the same realisation of $\tau' \cdot G(0,1)$ is used for all the variances of an individual. The reason for this is to allow an individual's total variance $\sigma = \left(\sum_{i=1}^{n} \sigma_i^2\right)^{1/2}$ to undergo significant change[4]. A point that will become relevant when comparing ESs to EP is that the variance is mutated by

[2] For $i = j$ there are no angles of rotation as $c_{ii} = \sigma_i^2$. It follows from this fact, combined with the knowledge that the covariance matrix is symmetric, that there are only $n \cdot (n-1)/2$ angles.

[3] A matrix M is said to be positive definite if $z^T M z$ is positive, for any non-zero column vector z. This is required so that all of the probabilities in the PD are non-zero.

[4] There is also the condition that the variances do not become arbitrarily small, that is $\sigma_i' = \epsilon$ if σ_i' drops below some predefined value ϵ.

multiplying it by exponentials of Gaussian distributions. This is to ensure that the variances have positive values, mutations are neutral (the expected adapted value is equal to the current value) and small modifications occur more often than larger ones.

The entire motive behind this complicated scheme is to enable the angles and variance of each mutation to be self-adaptive. This enables the PD of each individual to take its own form, examples of which may be seen in Figure 6.1.

6.1.2 Recombination

The other important aspect of ESs is recombination. Similar to crossover, recombination typically takes two parents to produce a child[5]. There are two traditional types of recombination, namely *discrete* and *intermediate*. In the discrete case each variable of the child has a fifty percent chance of coming from either parent. Intermediate recombination, on the other hand, assigns each variable the average of its parents' variables. Since each individual in an ES is a triple vector $(\vec{s}, \vec{\alpha}, \vec{\sigma})$, all of these variables may undergo recombination [6].

It may be noted that there are more recent and sophisticated versions of ESs not examined here. A popular type of ES is known as *Covariance Matrix Adaptation Evolution Strategy (CMA-ES)* [76] and has proven to be very effective [74].

The benefit of recombination for ESs is debatable, just as the benefit of crossover for GAs is dubious. Unlike in GAs, recombination is thought to be a secondary operator to mutation [157, 161]. This is even more so for EP, which does not use recombination at all.

6.2 Evolutionary Programming

David B Fogel took over the reigns of EP research thirty years after his father, Lawrence J Fogel, did his initial work on EP. In the younger Fogel's PhD thesis EP was applied to continuous search spaces for the first time [6]. He developed two main types of EP with their own unique mutation operators: the *standard EP*, with no self-adaptive mechanism, and the *meta-EP*, with variances (and covariances) used for self-adaptation.

6.2.1 Standard EP

In the standard EP the components of each individual are mutated according to the formula

$$s_i' = s_i + \sqrt{\gamma_i - f(\vec{s}) \cdot \beta_i} \cdot G_i(0,1),$$

where γ_i and $\beta_i > 0$ are parameters[6] and $G_i(0,1)$ is a random Gaussian variable sampled anew for each possible value of the counter i. Effectively this results in individuals of higher fitness having smaller step magnitudes.

The reason why such a scheme might be beneficial can be seen using PFL analysis. As argued in Section 5.3.3, due to selection there is a correlation between the density of individuals and their fitness. A consequence of this relationship is that it is desirable to have smaller step magnitudes

[5] If two parents are used, then recombination is called *sexual* and if more are used, then it is called *panmitic*. Although it is not uncommon to have panmitic recombination, sexual recombination is considered in this thesis because it is slightly simpler and its analysis is easily extended to panmitic recombination.

[6] The parameters must be tuned for each problem to appropriate values, which may be very difficult in practice [6].

for fitter individuals, which tend to be in higher density regions (for the full argument, refer to Section 5.3.3). Thus, standard EP mutation is beneficial.

6.2.2 Meta-EP

Meta-EP essentially has the same form as an ES. In its more basic form only the variances are adapted, in which case the update equations may be written as

$$\begin{aligned} s_i' &= s_i + \sqrt{\nu_i} \cdot G_i(0,1) \qquad \text{and} \\ \nu_i' &= \nu_i + \sqrt{\zeta \nu_i} \cdot G_i(0,1), \end{aligned}$$

where ν is equivalent to the variance in ES and ζ is a control parameter (the more complicated form also has covariances which adapt). There is only one significant difference between meta-EP and ESs, namely that the variance parameter for meta-EP ν is adapted additively, whereas for ES this is done multiplicatively via an exponential. The result is that the meta-EP variance is not guaranteed to be positive and is biased toward smaller values[7]. A consequence of this is that an ES is expected to be slightly better than meta-EP, although they should exhibit similar performance. The reason for their success is explained by *effective fitness*, discussed in the following section.

6.3 Effective Fitness

In its traditional form, natural selection states that fitter individuals are those which have a greater probability of survival. A more modern notion [131], sometimes referred to as *differential reproduction*, is that natural selection is "a process in nature in which organisms possessing certain genotypic characteristics that make them better adjusted to an environment tend to survive, reproduce, increase in number or frequency, and therefore, are able to transmit and perpetuate their essential genotypic qualities to succeeding generations." The key difference is the phrase "transmit and perpetuate," which is related to the notion of *effective fitness*. Effective fitness refers to whether an individual is able to produce fit offspring, not just whether it is capable of surviving selection [165]. Of course it is necessary for an individual to survive selection in order to produce offspring, but it is not sufficient to guarantee that the offspring will be fit.

Consider two individuals in the current population of an ES which have identical positions and angles, but have different variances. Assume that the variances of the one individual is appropriate for the problem, but that the variances of the other is too small. In this case both individuals have the exact same probability of surviving selection and are equally fit in this sense. However, it is likely that the offspring of the individual with the appropriate variances will be fitter and therefore this individual is of higher effective fitness.

The self-adaption of ESs and meta-EP enables the variances and angles to be adapted in order to improve the effective fitness of individuals. This is an optimisation problem in itself[8], hence

[7] To demostrate this consider the update equation $\nu_i' = \nu_i + \sqrt{\zeta \nu_i} \cdot G_i(0,1)$, with $\zeta = 1$ and $\nu_i = 3$, over two iterations. If $G_i(0,1) = +1$ for the first iteration and then $G_i(0,1) = -1$ for the second iteration, then $\nu_i' = 3 + \sqrt{3}$ and $\nu_i'' = 3 + \sqrt{3} - \sqrt{3 + \sqrt{3}} \approx 2.56 < 3$; whereas if $G_i(0,1) = -1$ for the first iteration and then $G_i(0,1) = +1$ for the second iteration, then $\nu_i' = 3 - \sqrt{3}$ and $\nu_i'' = 3 - \sqrt{3} + \sqrt{3 - \sqrt{3}} \approx 2.39 < 2.56 < 3$.

[8] There is, in fact, an infinite regress of optimisation problems: the parameters controlling the original problem may be controlled by other parameters, which in turn may be controlled by other parameters, which in turn may be controlled by other parameters, *etc.*

mutation and selection are used to optimise these values. Returning to the above example, the offspring of the individual with the small variances are likely to die out, which leaves the offspring of the other individual to spread throughout the population. Thus, the variances are optimised through selection. Combined with mutation, this process leads to appropriate variances and angles spreading through the population, thereby improving effective fitness and progressing the search.

6.4 Chapter summary

ESs and EP are very similar with both emphasising mutation as the predominant reproduction operator. This is especially the case for EP, which does not use any recombination. Instead these EAs rely mainly on the mutation operator. The standard EP has the mutation step size inversely proportional to the fitness of an individual, which improves the search due to density-fitness correlations. The other type of EP is known as meta-EP and is very similar to ESs, since they both use self-adaptive control parameters (*i.e.* variances and angles). Self-adaption is achieved though each individual having its own control parameters as variables, which are mutated every generation. Its success is substantiated by the notion of effective fitness, which argues that more optimal control parameters lead to fitter offspring and superior performance.

CHAPTER 7

Conclusion

Contents

The ideas behind EAs were investigated in this thesis, starting from their most fundamental concepts and gradually building the analysis up into a motivation for the sophisticated algorithms which are routinely used today. Although more complex EAs have been developed, they are still underpinned by the ideas here discussed. The aim in the thesis, as stated in the introduction, was to formalise these ideas in the form of a QM of EAs. This was achieved in four parts. First, the notion of Universal Darwinism was presented as the basis of EAs (see Chapter 1). Second, the historical development of EAs was traced and EAs were recognised as hill climbing algorithms that use all possible escape strategies to avoid premature convergence (see Chapter 2). Third, the PFL was presented as the foundation of a QM of EAs (see Chapter 4). This QM was used to define and analyse the prevalent views on *exploitation, exploration, intensity* and *diversity*[1] as well as explain why the principle operators of EAs, mutaiton and selection (presented in Chapter 3), are effective. Fourth, and finally, the PFL QM and the notion of effective fitness were used to explain the performance of GAs, ESs and EP (in Chapters 5 and 6). This demonstrated the success of the PFL as a basis of a QM of EAs, fulfilling the objective of the thesis.

The following section briefly contains an exposition on the differences in the philosophies behind GAs, ESs and EP, after which the notion of a QM is discussed. This is followed by a summary of the novel contributions made in the thesis and some ideas for possible future work.

7.1 Levels of Evolution

In biology, there are three main levels of evolution [54], each with its own fundamental unit: *genes*, *individuals* and *populations*. All of these units experience variation, inheritance and selection, and therefore have the characteristics required for Universal Darwinism and evolution (see Chapter 1). Each unit has its own unique mechanism for reproduction and selection, which

[1] A paper grew out of this chapter which has been provisionally accepted by the International Journal of Metaheuristics.

makes it similar to a type of EA. Genetic Algorithms, as the name suggests, are related to genes as they use crossover, mutation and select individuals that are encoded by multiple genes. ESs are similar to the level of the individual, since recombination and mutation[2] operators are employed and selection is applied to the individual. EP does not have recombination and is therefore most similar to populations. These similarities are summarised in Table 7.1.

Fundamental Unit	Reproduction	Selection	Algorithm
Gene	Crossover and mutation	Fit individuals' genes	GA
Individual	Recombination and mutation	Fit individuals	ES
Population	Mutation	Fit populations	EP

TABLE 7.1: *The different levels of evolution and their associated algorithms.*

The biological association of each EA leads to a different philosophy and a different type of algorithm. These philosophies are QMs, quite literally, in that they are trying to model qualitative characteristics of a level of biological evolution.

Ultimately, all biological evolution is reducible to the level of the gene, a view most famously popularised by the book *The selfish gene* [43]. This may have been a contributing factor to the popularity of GAs, as they most closely model how evolution occurs in nature. However, there is no reason to think *a priori* that if an algorithm's QM better models nature then it will be a more successful metaheuristic. These philosophical QMs cannot substantiate which type of EA is superior.

7.2 Scientific testing and facetwise models

It is likely that for each EA there are certain problems for which it will exhibit superior performance (in fact, this is guaranteed by the NFL theorems). However, algorithmic performance is not the only consideration that should be taken into account when deciding which algorithm to use [12]. Other considerations include [168, p.60], "the development cost, maintainability, ease of use, flexibility (wide applicability) and simplicity." These factors, and in particular simplicity, may be used to determine which EA is preferable for use in a specific context.

There is a strong argument that if an additional operator is not expected to increase algorithmic performance, then it should not be used (at least initially). The reasons for this is that an additional operator slows down the runtime of an algorithm, takes longer to code, increases the number of potential errors in the code, increases the conceptual complexity of the algorithm and increases the number of parameters to adjust. As all of these costs come at no expected benefit, it is irrational to include an additional operator in an algorithm initially. However, it may be useful to include an additional operator if the performance of the initial algorithm is unsatisfactory, since there will be problems for which the additional operator is beneficial (according to the NFL theorem).[3]

Mutation is the principle reproductive operator of EAs (see Section 4.6), which makes recombination (crossover) an additional operator. Generally, recombination is not expected to increase algorithmic performance (see Section 5.4) and therefore it should not (initially) be used. This

[2] Mutation does occur in individuals, such as humans [50].

[3] Yet even this may not be a good enough reason to use an additional operator. Alternative algorithms, which are completely unrelated to the initial algorithm, may have significantly better performance. Therefore, alternative algorithms should be considered before incorporating an additional operator, unless doing so would require significantly more coding.

implies that mutation-only EAs are generally preferable. Specifically, mutation-only GAs are generally preferable to GAs with crossover and ESs are generally better without recombination.

Naturally there will be cases when additional operators improve performance. Thus, the purpose of EA research should be to determine when these additional operators are beneficial. This is the approach promoted by Hooker [84] who advocates *scientific testing* over competitive testing. A resonant view is expressed by Goldberg (who is arguably the most influential person in the field of evolutionary computation [36]), as mentioned in Chapter 1. He outlines a spectrum of approaches for analysing and modelling algorithms: from rough intuitive models on the far left, to exact technical models on the far right, shown in Figure 7.1.

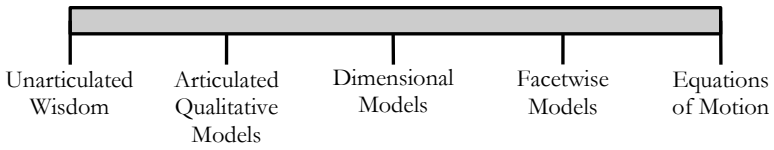


FIGURE 7.1: *The modelling spectrum of Goldberg [69].*

Due to the lack of a soluble set of equations of motion for EAs, Goldberg argues that the next most promising approach is *facetwise modelling*. Facetwise modelling turns out to be the same as the approach advocated by Hooker, in that it scientifically analyses facets (characteristics) of algorithms. These facetwise models may then be mathematically combined, or *patch-quilt integrated*, into more extensive *dimensional models*. Together, facetwise and dimensional models give a good idea as to how certain algorithms behave in certain problem classes. This information can be utilised to determine when additional operators are beneficial and should be used.

7.3 Qualitative Models

QMs are one notch down from dimensional models on Goldberg's spectrum. This is misleading as it implies that QMs are not equations of motion, facetwise models or dimensional models, when QMs are, in fact, qualitative versions of these models. For example, Universal Darwinism is a QM which describes the equations of motion — the governing mechanics — of EAs. Intensity, diversity, hill climbing and escape strategies are facetwise QMs in that they describe a facet of an EA. Terminology, such as exploitation and exploration, integrate many facets of an EA into larger, more sophisticated, dimensional models[4]. The only difference between QMs and Goldberg's other models is that the other models are quantitative[5], whereas QMs are qualitative[6]. Perhaps a more appropriate spectrum is given in Figure 7.2.

The most challenging question that QMs have to answer to is why they should be taken seriously? If they are not quantifiable, then they are not falsifiable and should therefore be dismissed out of hand. The answer to this charge is that QMs may communicate important ideas which are

[4]Calling a QM a dimensional model is a slight abuse of terminology. Goldberg intended dimensional models to refer to the use of mathematical dimensional analysis for combining facetwise models. However, the point of dimensional models is to combine facetwise models so as to yield new insights, and in this spirit the notion of a dimensional model is extended to QMs.

[5]Quantitative models are necessarily mathematically expressed and may be derived via deduction (using mathematics) or induction (by means of empirical experimentation).

[6]According to Goldberg, QMs range from "verbal descriptions of mechanism to pictorial or graphical representations of processes or relationships" [69].

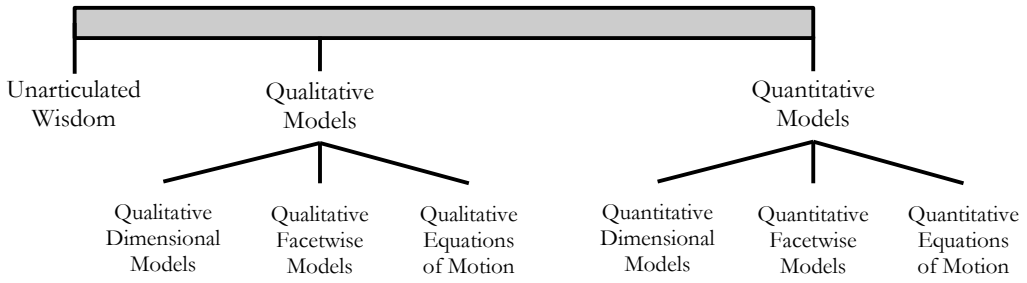


FIGURE 7.2: *An adaption of Goldberg's modelling spectrum.*

not effectively communicable in a quantitative form. These ideas may be valuable and therefore should not be dismissed.

A powerful example of a QM being used to communicate ideas is the fitness landscape (see Section 4.2.2). Introduced by Wright [183] to explain his theory of population genetics, the image of his fitness landscape communicated his ideas more effectively than his mathematics. The notion of a fitness landscape became "the crucial key to the understanding of evolution" [149], although it did nothing more than make visual Wright's mathematical ideas. Likewise, the fitness landscape has become a key concept for understanding EAs.

It may be argued that even if QMs are useful it would be better if they could be made quantitative. To address this argument it is instructive to consider the terms exploitation and exploration. As stated in [41], in order to assess (and potentially control) exploitation and exploration they must be measured, that is, made quantitative. Although some methods of measurement have been proposed in the literature [105, 41], how to do this is still an open question. Črepinšek *et al.* [41, p.8] remark that, "Intrinsic to this problem is that we need to know how these two phases [(exploitation and exploration)] are identified. If in each process both phases can be clearly identified, then some direct measures can be invented. Currently, indirect measures for exploration and exploitation are mostly used." The PFL QM goes some of the way to identifying the facets of exploitation and exploration (see Section 4.3). As the PFL is a hypothetical construct that cannot be calculated with certainty, it cannot be used as a direct measure of exploitation and exploration. However, the identified facets may serve as "indirect measures." For example, diversity may be used in combination with a distance metric as a measure [105, 41]. The $1/5^{th}$-success rule [20] may be interpreted as a measure of whether the search is local or global to measure exploitation and exploration. Which facets make for the best measure is likely to be problem-dependent[7] [41, p.6]. Therefore, if exploitation and exploration were quantitative, then they would be problem-dependent and not universal. However, if they are qualitative, then they can be universal and have the flexibility to be adapted to any problem (using appropriate quantitative measures[8]). Thus there is an advantage for certain ideas[9] being qualitative as opposed to quantitative.

It may be asked why QMs, such as the Schema Theorem and BBH, have such a dubious track record (see Section 5.3.2)? The answer is that they were branded as equations of motion, when they are actually facetwise models. As was discussed in Section 5.4, the Schema Theorem

[7] Some facetwise terms are also problem-dependent, such as diversity [41, p.8].

[8] The question of which quantitative measure best captures the qualitative notions of exploitation and exploration then has to be answered separately for each problem. This would leave the universal notions of exploitation and exploration as qualitative, with problem-specific quantitatively measures.

[9] Of course, if qualitative ideas can be made quantitative and remain universal, then they should be. The point, however, is that this is not possible for all ideas.

and BBH simply describe two facets of GAs, namely that "crossover causes BBs to increase in frequency" and that a "maximum solution may be decomposed into BBs" (for non-deceptive problems). Likewise, the EPP and GRH recognise the facets[10] that "evolutionary progress = progress gain – progress loss" and "crossover causes a decrease in progress loss" (for convex problems). These theories, principles and hypotheses reflect a desire to explain why EAs are generally preferable, but necessarily fail because the statement that they are trying to explain is false (*i.e.* EAs are not generally preferable, see Section 5.4). An equation of motion for an EA can at most explain why an EA is expected to converge toward a maximum, and this is achieved by Universal Darwinism[11]. All other QMs of EAs are therefore either facetwise or dimensional.

The facetwise approach is supported in the recent book on predictions by Nate Silver entitled *The signal and the noise* [159]. In it he suggests that there are roughly two types of people who make predictions, hedgehogs and foxes. "Hedgehogs, Silver says, are those who believe in governing principles about the world that behave as though they were physical laws. Foxes, by contrast, are scrappy creatures who believe in a plethora of little ideas and in taking a multitude of approaches toward a problem" [178]. Silver makes the convincing argument throughout the book (covering topics such as politics, sport, finance, earthquakes and terrorism) that foxes make better predictions than hedgehogs, that a facetwise approach is preferable to trying to discover the equations of motion.

The PFL provides the foundations of a QM upon which facets may be appended to construct a comprehensive understanding of EAs. It was first used as a basis for the definitions of the dimensional (multi-faceted) terms of exploitation and exploration (see Section 4.3). Several facets of these notions, identified in a literature review, were then shown to be deducible from the PFL definitions. This unified all of the facets into one coherent framework. The PFL in itself is a substantial contribution in that it, like Wright's fitness landscape, makes visual certain ideas. On top of this, it has been used to analyse continuity (and long-range correlations) and is closely related to meta-modelling (see Sections 4.2.1 and 4.2.3, respectively). Next, the PFL was extended to the IPD, which is a tool for acertaining whether certain facets are beneficial or not. It was used to motivate the use of mutation and selection (see Section 4.6), and later GAs and EP (see Sections 5.3.3 and 6.2, respectively). The strengths of the PFL and IPD were most evident when compared to the Schema Theorem, BBH, EPP and GRH — it was clear the facetwise approach is far more fruitful than naive attempts to explain why GAs are better.

In summary, models of EAs have been investigated in this thesis. It has been motivated that QMs are useful and the PFL has beeen proposed as the basis for a QM of EAs. Finally, it has been argued that facetwise models, both qualitative and quantitative, are the appropriate type of model for EA research.

7.4 Novel contributions of this thesis

With the literature on EAs being as vast as it is, it is difficult to know with certainty that an idea or application is novel. However, to the best of the author's knowledge, the following contributions made in this thesis are novel. They are listed according to the chapters in which they were presented.

[10] The statement that evolutionary progress is separable is disputed by Vose and Liepins's mathematical results (see Section 5.4).

[11] Note that since Universal Darwinism only depends on continuity, it is universal over the class of problems to which metaheuristics are applicable [62, p.2129]. Therefore it holds for any EA.

1. **Introduction**
   - An explanation of how Universal Darwinism implies that EAs should converge toward the maximum solution
2. **The Development of Evolutionary Algorithms**
   - A description of EAs as hill climbing algorithms with all possible escape strategies
   - The correction of Theorem 2.4.2
   - The correction of Lemma 2.5.1
   - The construction of Theorem 2.5.3
3. **Principal Operators of Evolutionary Algorithms**
   - The motivation of the shape of mutation distributions via Universal Darwinism
   - An approximation of the binary mutation distribution with a continuous distribution
4. **The Probable Fitness Landscape**
   - A literature review in which six prevalent views on exploitation and exploration were identified
   - The discovery that the terms exploration and diversity are more frequently used than exploitation and exploration
   - The notion of the PFL
   - A comparison of the PFL to meta-models and traditional fitness landscapes
   - The identification of the MAX-3-SAT paper as an application of the PFL
   - The definitions of exploitation and exploration based on the PFL
   - The deduction of the six prevalent views on exploitation and exploration from the PFL definitions
   - An explanation of the benefits of exploitation, exploration and diversity
   - The extention of the PFL to the IPD
   - A demonstration of how the IPD motivates the use of mutation and selection
5. **Genetic Algorithms**
   - The observation that crossover maximises the distance between children without altering their PDs
   - The derivation of the genetic drift equation
   - A comparison of the continuous mutation and crossover PDs
   - An analysis of the ellipsoidal crossover PD
   - An examination of the Schema Theorem, BBH, GRH, EPP and PFL argument using the notions of validity and soundness
6. **Evolution Strategies and Evolutionary Programming**
   - An explanation of ESs and meta-EP using the notion of effective fitness
   - An explanation of the standard EP appealing to PFL
7. **Conclusion**
   - A defence of QMs

7.5 Possible future work

A number of directions of future research are suggested by the contents of this thesis, as listed below.

1. **To conduct a deeper literature review of the QMs in EAs**
   Specifically, to uncover more facets of exploitation and exploration and attempt to deduce them from the PFL definitions.

2. **To develop explicit methods for calculating the PFL and IPD**
   Although this is necessarily a futile exercise (the PFL and IPD are inherently uncertain), this endeavour might be useful for communicating the ideas of researchers and may lead to new algorithms.

3. **To incorporate the expected fitness decreasing away from candidate solutions into meta-models**
   Traditionally meta-models use all previously generated candidate solutions to approximate the fitness function, but as the search progresses and the number of previously candidate solutions increases, this process becomes very slow (see Section 4.2.1). The process could be speeded up by not including unfit candidate solutions in the calculations, since the information acquired from these candidate solutions is the least valuable. Their omission could be countered by incorporating the expected fitness decreasing away from candidate solutions in the fitness function approximation.

4. **To analyse the PFL as a method for investigating long-range correlations**
   It is not clear what may be concluded from the fact that a problem obeys the PFL principles (see Section 4.2.3). The observance of the PFL principles suggest a type of continuity, but it is uncertain how this relates to long-range correlations.

5. **To analyse other operators and algorithms using the PFL argument**
   For example, to apply the PFL argument to operators, such as mean-centered crossover, or algorithms, for instance: Differential Evolution, Scatter Search, Estimation of Distribution Algorithm, Ant Colony Optimisation and Particle Swarm Optimisation.

6. **To compare mutation and crossover PDFs**
   Although a superficial comparison between the mutation and crossover PDFs has been made (see Section 5.2.4), a more thorough analysis could yield interesting conclusions.

7. **To develop a real GA that simulates binary GAs**
   A GA which uses real numbers instead of binary numbers could be used to investigate certain facets of binary GAs. A test for whether a real GA accurately simulates a binary GA would be to simulate both algorithms on a number of problems and determine whether their runtime performance is similar. If an accurate real GA is developed, then facets of the binary GA could be investigated by altering the real GA and observing the results. Possible alterations include: not rotating the crossover step vectors by 180 degrees (see Section 5.2.1); making the crossover step magnitude inversely proportional to parent's separation distance (see Section 5.2.4); and changing the elliptical eccentricity of the crossover step vector (see Section 5.2.5).

8. **To discover and analyse more facets of EAs**
   This is the suggested general direction for future research in EAs, as motivated in the conclusion.

References

[1] Alam MS, Kabir MWU & Islam MM, 2010, *On the performance of recurring multi-stage evolutionary algorithm for continuous function optimization*, Proceedings of the 13th International Conference on Computation and Information Technology, Dhaka, pp. 63–68.

[2] Alba E & Dorronsoro B, 2005, *The exploration/exploitation tradeoff in dynamic cellular genetic algorithms*, IEEE Transactions on Evolutionary Computation, **9(2)**, pp.126–142.

[3] Anderson RL, 1953, *Recent advances in finding best operating conditions*, Journal of the American Statistical Association, **48**, pp.789–798.

[4] Auger A & Doerr B, 2011, *Theory of randomized search heuristics: Foundations and recent developments*, World Scientific Publishing Company, Hackensack (NJ).

[5] Baba N, 1981, *Convergence of a random optimization method for constrained optimization problems*, Journal of Optimization Theory and Applications, **33(4)**, pp.451–461.

[6] Bäck T, 1996, *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press, Oxford.

[7] Bäck T & Hoffmeister F, 1991, *Extended selection mechanisms in genetic algorithms*, Proceedings of the 4th International Conference on Genetic Algorithms and Their Application, San Diego (CA), pp. 92–99.

[8] Bäck T, Rudolph G & Schwefel H, 1993, *Evolutionary programming and evolution strategies: Similarities and differences*, Proceedings of the 2nd Annual Conference on Evolutionary Programming, San Francisco (CA), pp. 11–22.

[9] Baker JE, 1987, *Reducing bias and inefficiency in the selection algorithm*, Proceedings of the 2nd International Conference on Genetic Algorithms and Their Application, Cambridge (MA), pp. 14–21.

[10] Bank of England Quarterly Bulletin, 1998, *The inflation report projections: Understanding the fan chart*, (Unpublished) Technical Report 1, Bank of England, London.

[11] Baños R, Gil C, Paechter B & Ortega J, 2006, *Parallelization of population-based multi-objective meta-heuristics: An empirical study*, Applied Mathematical Modelling, **30(7)**, pp.578–592.

[12] Barr RS, Golden BL, Kelly JP, Resende MGC & Stewart WR, 1995, *Designing and reporting on computational experiments with heuristic methods*, Journal of Heuristics, **1(1)**, pp.9–32.

[13] Baxter J, 1981, *Local optima avoidance in depot location*, Journal of the Operational Research Society, **38**, pp.815–819.

[14] Beasley D, Bull DR & Martin RR, 1993, *An overview of genetic algorithms: Part 2, Research topics*, University Computing, **15(4)**, pp.170–181.

[15] Beyer H, 1997, *An alternative explanation for the manner in which genetic algorithms operate*, BioSystems, **41**, pp.1–15.

[16] Beyer H, 1998, *On the "explorative power" of ES EP-like algorithms*, Proceedings of the Evolutionary Programming VII: 7th Annual Conference on Evolutionary Programming, Berlin, pp. 323–334.

[17] Beyer H, 2001, *The theory of evolution strategies*, Springer Verlag, Heidelberg.

[18] Beyer H, 1994, *Toward a theory of evolution strategies: On the benefits of sex — The $(\mu/\mu, \lambda)$-theory*, Evolutionary Computation, **2(4)**, pp.381–407.

[19] Beyer H, 1994, *Toward a theory of evolution strategies: The (μ, λ)-Theory*, Evolutionary Computation, **2(4)**, pp.381–407.

[20] Beyer H & Schwefel H, 2002, *Evolution strategies — A comprehensive introduction*, Natural Computing, **1(1)**, May, pp.3–52.

[21] Blickle T & Thiele L, 1995, *A comparison of selection schemes used in genetic algorithms*, (Unpublished) Technical Report, Computer Engineering and Communications Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

[22] Blum C & Roli A, 2003, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, ACM Computing Surveys, **35(3)**, pp.268–308.

[23] Borowski N, 1961, *A comparison of three random search methods*, Masters Thesis, University of British Columbia, Vancouver.

[24] Bosman PAN & Thierens D, 2003, *The balance between proximity and diversity in multiobjective evolutionary algorithms*, IEEE Transactions on Evolutionary Computation, **7(2)**, pp.174–188.

[25] Brooks SH, 1958, *A discussion of random methods for seeking maxima*, Operations Research, **6**, pp.244–251.

[26] Burke E, Curtois T, Hyde M, Kendall G, Ochoa G, Petrovic S, Vázquez-Rodríguez JA & Gendreau M, 2010, *Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms*, Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, pp. 1–8.

[27] Campbell DT, 1960, *Blind variation and selective retention in creative thought as in other knowledge processes*, Psychological Review, **67(6)**, pp.380–400.

[28] Cavicchio DJ, 1970, *Adaptive search using simulated evolution*, PhD Thesis, University of Michigan, Ann Arbor (MI).

[29] Chaiyaratana N & Zalzala AMS, 1997, *Recent developments in evolutionary and genetic algorithms: Theory and applications*, Proceedings of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, Glasgow, pp. 270–277.

[30] Chen F, Sun X & Wei D, 2011, *Inertia weight particle swarm optimization with Boltzmann exploration*, Proceedings of the 7th International Conference on Computational Intelligence and Security, Sanya, pp. 90–95.

[31] Chen F, Sun X, Wei D & Tang Y, 2011, *Tradeoff strategy between exploration and exploitation for PSO*, Proceedings of the 7th International Conference on Natural Computation, Shanghai, pp. 1216–1222.

[32] Chen J, Xin B, Peng Z, Dou L & Zhang J, 2009, *Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization*, IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans, **39(3)**, pp.680–691.

[33] Chen T, He J, Sun G, Chen G & Yao X, 2009, *A new approach for analyzing average time complexity of population-based evolutionary algorithms on unimodal problems*, IEEE Transactions on Systems, Man, and Cybernetics — Part B, **39(5)**, pp.1092–1106.

[34] Chiong R, Weise T & Michalewicz Z, 2011, *Variants of evolutionary algorithms for real-world applications*, Springer-Verlag, Berlin.

[35] Choit MD, 1970, *Optimized relative step size random search*, Masters Thesis, University of British Columbia, Vancouver.

[36] Citeseerx, 2013, *Most Cited Computer Science Citations*, This list is generated from documents in the CiteSeerx database as of January 17, 2013. [Online; Accessed July 13, 2013], url: `http://citeseerx.ist.psu.edu/stats/citations`.

[37] Cohen G, Litsyn S & Zemor G, 1996, *On the traveling salesman problem in binary hamming spaces*, IEEE Transactions on Information Theory, **42(4)**, pp.1274–1276.

[38] Cohen PR, 1995, *Empirical methods for artificial intelligence*, MIT Press, Cambridge (MA).

[39] Copeland BJ, 2004, *The essential Turing*, Oxford University Press, Oxford.

[40] Couceiro MS, Rocha RP, Ferreira NMF & Machado JAT, 2012, *Introducing the fractional-order Darwinian PSO*, Signal, Image and Video Processing, **6(3)**, pp.343–350.

[41] Črepinšek M, Liu S & Mernik M, 2013, *Exploration and exploitation in evolutionary algorithms: A survey*, ACM Computing Surveys, **45(3)**, pp.35:1–35:33.

[42] Darwin C, 1872, *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*, J Murray, London.

[43] Dawkins R, 1989, *The selfish gene*, Oxford University Press, Oxford.

[44] Dawkins R, 1983, *Universal Darwinism*, pp. 403–425 in Bendall DS (Ed), *Evolution from molecules to men*, Cambridge University Press, Cambridge.

[45] De la Maza M & Tidor B, 1993, *An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection*, Proceedings of the 5th International Conference on Genetic Algorithms, San Francisco (CA), pp. 124–131.

[46] Deb K & Agrawal RB, 1994, *Simulated binary crossover for continuous search space*, (Unpublished) Technical Report, Departement of Mechanical Enginering, Indian Institute of Technology, Kanpur.

[47] Deb K & Beyer H, 2001, *Self-adaptive genetic algorithms with simulated binary crossover*, Evolutionary Computation, **9(2)**, pp.197–221.

[48] Dekkers A & Aarts E, 1991, *Global optimization and simulated annealing*, Mathematical Programming, **50**, pp.367–393.

[49] Dorigo M & Stützle T, 2003, *The ant colony optimization metaheuristic: Algorithms, applications and advances*, pp. 251–286 in Glover FW & Kochenberger GA (Eds), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[50] Drake JW, Charlesworth B, Charlesworth D & Crow JF, 1998, *Rates of spontaneous mutation*, Genetics, **148(4)**, pp.1667–1686.

[51] Dréo J, Pétrowski A, Siarry P & Taillard E, 2006, *Metaheuristics for hard optimization*, Springer-Verlag, Heidelberg.

[52] Droste S, Jansen T & Wegener I, 1997, *Optimization with randomized search heuristics — The (A)NFL theorem, realistic scenarios, and difficult functions*, Theoretical Computer Science, **287**, pp.131–144.

[53] Eberhart RC & Shi Y, 1998, *Comparison between genetic algorithms and particle swarm optimization*, pp. 611–616 in Porto VW, Saravanan N, Waagen D & Eiben AE (Eds), *Evolutionary programming VII*, Springer, Berlin.

[54] Eiben AE & Schippers CA, 1998, *On evolutionary exploration and exploitation*, Fundamenta Informaticae, **35(1–4)**, pp.35–50.

[55] Emmerich M, Giotis A, Özdemir M, Bäck T & Giannakoglou K, 2002, *Metamodel-assisted evolution strategies*, Proceedings of the 7th International Conference on Parallel Problem Solving from Nature (PPSN VII), London, pp. 361–370.

[56] Feo TA & Resende MGC, 1989, *A probabilistic heuristic for a computationally difficult set covering problem*, Operations Research Letters, **8(2)**, pp.67–71.

[57] Feo TA & Resende MGC, 1995, *Greedy randomized adaptive search procedures*, Journal of Global Optimization, **6**, pp.109–133.

[58] Fogel DB & Atmar JW, 1990, *Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems*, Biological Cybernetics, **63**, pp.111–114.

[59] Fogel DB & Chellapilla K, 1998, *Revisiting evolutionary programming*, Proceedings of the SPIE, Applications and Science of Computational Intelligence, Orlando (FL), pp. 2–11.

[60] Fogel DB & Ghozeil A, 1997, *Schema processing under proportional selection in the presence of random effects*, IEEE Transactions on Evolutionary Computation, **1(4)**, pp.290–293.

[61] García-Martínez C, Lozano M, Herrera F, Molina D & Sánchez AM, 2008, *Global and local real-coded genetic algorithms based on parent-centric crossover operators*, European Journal of Operational Research, **185(3)**, pp.1088–1113.

[62] García-Martínez C, Rodriguez FJ & Lozano M, 2012, *Arbitrary function optimisation with metaheuristics*, Soft Computing, **16(12)**, pp.2115–2133.

[63] Geman S & Geman D, 1984, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **6(6)**, pp.721–741.

[64] Gendreau M, 2003, *An introduction to tabu search*, pp. 185–218 in Glover F & Kochenberger GA (Eds), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[65] Glover F, 1986, *Future paths for integer programming and links to artificial intelligence*, Computers and Operations Research, **13(5)**, pp.533–549.

[66] Glover F & Laguna M, 1997, *Tabu search*, Kluwer Academic Publishers, Norwell (MA).

[67] Glover F, Laguna M & Marti R, 2003, *Scatter search and path relinking: Advances and applications*, pp. 1–36 in Glover FW & Kochenberger GA (Eds), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[68] Goldberg DE, 1989, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Longman Publishing Co., Inc., Boston (MA).

[69] Goldberg DE, 2002, *The design of innovation: Lessons from and for competent genetic algorithms*, Kluwer Academic Publishers, Norwell (MA).

[70] Goldberg DE & Deb K, 1991, *A comparative analysis of selection schemes used in genetic algorithms*, pp. 69–93 in Rawlins GJE (Ed), *Foundations of genetic algorithms*, Morgan Kaufmann, San Francisco (CA).

[71] Granville V, Krivanek M & Rasson J, 1994, *Simulated annealing: A proof of convergence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **16(6)**, pp.652–656.

[72] Grefenstette J, 1992, *Deception considered harmful*, Proceedings of the Foundations of Genetic Algorithms 2, Vail (CO), pp. 75–91.

[73] Hancock PJB, 1994, *An empirical comparison of selection methods in evolutionary algorithms*, Proceedings of the Selected Papers from AISB Workshop on Evolutionary Computing, London, pp. 80–94.

[74] Hansen N, Auger A, Ros R, Finck S & Pošík P, 2010, *Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009*, Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, Portland (OR), pp. 1689–1696.

[75] Hansen N, Gemperle F, Auger A & Koumoutsakos P, 2006, *When do heavy-tail distributions help?*, Proceedings of the 9th International Conference on Parallel Problem Solving from Nature, Reykjavik, pp. 62–71.

[76] Hansen N & Ostermeier A, 1996, *Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation*, Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, pp. 312–317.

[77] Hansen P & Mladenović N, 2001, *Variable neighborhood search: Principles and applications*, European Journal of Operational Research, **130(3)**, pp.449–467.

[78] Hansheng L & Lishan K, 1999, *Balance between exploration and exploitation in genetic search*, Wuhan University Journal of Natural Sciences, **4(1)**, pp.28–32.

[79] Harman M, Mansouri SA & Zhang Y, 2009, *Search based software engineering: A comprehensive analysis and review of trends techniques and applications*, (Unpublished) Technical Report, King's College, London.

[80] Hastings WK, 1970, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, **57(1)**, pp.97–109.

[81] Henderson D, Jacobson SH & Johnson AW, 2003, *The theory and practice of simulated annealing*, pp. 287–320 in Glover F & Kochenberger GA (Eds), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[82] Herrera F, Lozano M & Verdegay JL, 1998, *Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis*, Artificial Intelligence Review, **12(4)**, pp.265–319.

[83] Holland JH, 1975, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor (MI).

[84] Hooker JN, 1995, *Testing heuristics: We have it all wrong*, Journal of Heuristics, **1(1)**, pp.33–42.

[85] Hoos HH & Stützle T, 2004, *Stochastic local search: Foundations and applications*, Morgan Kaufmann, San Francisco (CA).

[86] Internet encyclopaedia of philosophy T, 2013, *Validity and soundness*, [Online; Accessed May 27, 2013], URL: `http://www.iep.utm.edu/val-snd/`.

[87] Iorio A & Li X, 2006, *Rotationally invariant crossover operators in evolutionary multi-objective optimization*, pp. 310–317 in Wang T, Li X, Chen S, Wang X, Abbass H, Iba H, Chen G & Yao X (Eds), *Simulated evolution and learning*, Springer, Berlin.

[88] Ishibuchi H, Tsukamoto N & Nojima Y, 2008, *Evolutionary many-objective optimization: A short review*, Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, pp. 2419–2426.

[89] Jägersküpper J, 2003, *Analysis of a simple evolutionary algorithm for minimization in Euclidean spaces*, Proceedings of the 30th International Conference on Automata, Languages and Programming, Eindhoven, pp. 1068–1079.

[90] Jansen T & Wegener I, 2001, *On the utility of populations in evolutionary algorithms*, Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco (CA), pp. 1034–1041.

[91] Jin Y, 2005, *A comprehensive survey of fitness approximation in evolutionary computation*, Soft Computing, **9(1)**, pp.3–12.

[92] Johnson DS, 2002, *A theoretician's guide to the experimental analysis of algorithms*, Proceedings of the 5th and 6th DIMACS Implementation Challenges on Data Structures, Near Neighbor Searches, and Methodology, Providence (RI), pp. 215–250.

[93] Jong KAD, 1975, *An analysis of the behaviour of a class of genetic adaptive systems*, PhD Thesis, University of Michigan, Ann Arbor (MI).

[94] Karnopp DC, 1963, *Random search techniques for optimization problems*, Automatica, **1(2–3)**, pp.111–121.

[95] Khan K & Sahai A, 2012, *A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context*, International Journal of Intelligent Systems and Applications, **7**, pp.23–29.

[96] Kirkpatrick S, Gelatt CD & Vecchi MP, 1983, *Optimization by simulated annealing*, Science, **220(4598)**, pp.671–680.

[97] Kita H, Ono I & Kobayashi S, 1998, *Theoretical analysis of the unimodal normal distribution crossover for real-coded genetic algorithms*, Proceedings of the IEEE International Conference on Evolutionary Computation, World Congress on Computational Intelligence, Piscataway (NJ), pp. 529–534.

[98] Kleijnen JPC, 2009, *Kriging metamodeling in simulation: A review*, European Journal of Operational Research, **192(3)**, pp.707–716.

[99] Krzanowski R & Raper J, 2001, *Spatial evolutionary modeling*, Oxford University Press, New York (NY).

[100] Larrañaga P & Lozano JA, 2005, *Estimation of distribution algorithms: A new tool for evolutionary computation*, Spring-Verlag, Berlin.

[101] Lehre PK & Yao X, 2009, *On the impact of the mutation-selection balance on the runtime of evolutionary algorithms*, Proceedings of the 10th ACM SIGEVO workshop on Foundations of Genetic Algorithms, New York (NY), pp. 47–58.

[102] Leung Y, Gao Y & Xu Z, 1997, *Degree of population diversity — A perspective on premature convergence in genetic algorithms and its Markov chain analysis*, IEEE Transactions on Neural Networks, **8**, pp.1165–1176.

[103] Lewontin RC, 1970, *The units of selection*, Annual Review of Ecology and Systematics, **1**, pp.1–18.

[104] Linhares A & Yanasse HH, 2010, *Search intensity versus search diversity: A false trade off?*, Applied Intelligence, **32(3)**, pp.279–291.

[105] Liu S, Črepinšek M & Mernik M, 2012, *Analysis of VEGA and SPEA2 using exploration and exploitation measures*, Proceedings of the 5th International Conference on Bio-inspired Optimization Methods and Their Applications, Bohinj, pp. 97–100.

[106] Locatelli M, 2000, *Convergence of a simulated annealing algorithm for continuous global optimization*, Journal of Global Optimization, **18(3)**, pp.219–233.

[107] Lourenço HR, Martin OC & Stützle T, 2001, *A beginner's introduction to iterated local search*, Proceedings of the 4th Metaheuristics International Conference, Porto, pp. 1–6.

[108] Lourenço HR, Martin OC & Stützle T, 2003, *Iterated local search*, pp. 321–353 in Glover FW & Kochenberger GA (Eds), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[109] Luke S, 2012, *Essentials of metaheuristics*, Undergraduate Lecture Notes, George Mason University, Fairfax (VA).

[110] Luke S & Spector L, 1997, *A comparison of crossover and mutation in genetic programming*, Proceedings of the 2nd Annual Conference on Genetic Programming, San Francisco (CA), pp. 240–248.

[111] Luque G, Luna F & Alba E, 2012, *Enhanced parallel cooperative model for trajectory based metaheuristics: A scalability analysis*, Proceedings of the 26th IEEE International Parallel and Distributed Processing Symposium Workshops PhD Forum, Shanghai, pp. 656–660.

[112] Mahfoud SW, 1992, *Crowding and preselection revisited*, Proceedings of the 2nd Conference on Parallel Problem Solving from Nature, Brussels, pp. 27–36.

[113] Mahfoud SW, 1996, *Niching methods for genetic algorithms*, PhD Thesis, University of Illinois, Champaign (IL).

[114] Martí R, 2003, *Multi-start methods*, pp. 355–368 in Glover F (Ed), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[115] Martí R, Resende MGC & Ribeiro CC, 2013, *Multi-start methods for combinatorial optimization*, European Journal of Operational Research, **226(1)**, pp.1–8.

[116] Matai R, Singh S & Mittal ML, 2010, *Traveling salesman problem: An overview of applications, formulations, and solution approaches*, pp. 1–24 in Davendra D (Ed), *Traveling salesman problem, theory and applications*, InTech, Rijeka.

[117] Matyas J, 1968, *Das zufällige Optimierungsverfahren und seine Konvergenz*, Proceedings of the 5th International Analogue Computation Meeting, Brussels, pp. 540–544.

[118] Matyas J, 1965, *Random optimization*, Automation and Remote Control, **26(2)**, pp.244–251.

[119] Mendes A & Linhares A, 2004, *A multiple-population evolutionary approach to gate matrix layout*, International Journal of Systems Science, **35(1)**, pp.13–23.

[120] Mernik M, Liu S & Bryant BR, 2007, *Entropy-driven parameter control for evolutionary algorithms*, Informatica: An International Journal of Computing and Informatics, **31(1)**, pp.41–50.

[121] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH & Teller E, 1953, *Equation of state calculations by fast computing machines*, Journal of Chemical Physics, **21(6)**, pp.1087–1092.

[122] Mitchell M, 1998, *An introduction to genetic algorithms*, MIT Press, Cambridge (MA).

[123] Mladenović N & Hansen P, 1997, *Variable neighborhood search*, Computers and Operations Research, **24(11)**, pp.1097–1100.

[124] Mühlenbein H, 1997, *The equation for response to selection and its use for prediction*, Evolutionary Computation, **5(3)**, pp.303–346.

[125] Mutseniyeks VA & Rastrigin LA, 1964, *Extremal control of continuous multi-parameter systems by the method of random search*, Engineering Cybernetics, **1**, pp.82–90.

[126] Nakamichi Y & Arita T, 2004, *Diversity control in ant colony optimization*, Artificial Life and Robotics, **7(4)**, pp.198–204.

[127] Al-Naqi A, Erdogan AT, Arslan T & Mathieu Y, 2010, *Balancing exploration and exploitation in an adaptive three-dimensional cellular genetic algorithm via a probabilistic selection operator*, Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems, Anaheim (CA), pp. 258–264.

[128] Naudts B & Schippers A, 1999, *A motivated definition of exploitation and exploration*, Proceedings of the Genetic and Evolutionary Computation Conference, Orlando (FL), pp. 800–807.

[129] Ollion C & Doncieux S, 2011, *Why and how to measure exploration in behavioral space*, Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, New York (NY), pp. 267–274.

[130] Olsson AE, 2011, *Particle swarm optimization: Theory, techniques and applications*, Nova Science Pub Incorporated, New York (NY).

[131] Online B, 2013, *Natural selection*, [Online; Accessed July 1, 2013], url: `http://www.biology-online.org/dictionary/Natural_selection`.

[132] Ortiz-boyer D, Hervás-Martínez C & García-pedrajas N, 2005, *CIXL2 — A crossover operator for evolutionary algorithms based on population features*, Journal of Artificial Intelligence Research, **24**, pp.1–48.

[133] Pan X, Zhang J & Szeto KY, 2005, *Application of mutation-only genetic algorithm for the extraction of investment strategy in financial time series*, Proceedings of the International Conference on Neural Networks and Brain, Beijing, pp. 1682–1686.

[134] Papadimitriou CH & Steiglitz K, 1982, *Convergence properties of evolutionary algorithms*, Prentice-Hall, Inc., Upper Saddle River (NJ).

[135] Poli R, 2000, *Why the schema theorem is correct also in the presence of stochastic effects*, Proceedings of the Congress on Evolutionary Computation, San Diego (CA), pp. 487–492.

[136] Price KV, Storn RM & Lampinen JA, 2005, *Differential evolution, a practical approach to global optimization*, Spring-Verlag, Berlin.

[137] Prugel-Bennett A & Tayarani-Najaran M, 2012, *Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimization problem*, IEEE Transactions on Evolutionary Computation, **16(3)**, pp.319–338.

[138] Rajasekaran S, 1990, *On the convergence time of simulated annealing*, Research report MS-CIS-90-89, University of Pennsylvania, Department of Computer and Information Science.

[139] Rastrigin LA, 1963, *The convergence of the random search method in the extremal control of a many parameter system*, Automation and Remote Control, **24(10)**, pp.1337–1342.

[140] Rechenberg I, 1973, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog Verlag, Stuttgart.

[141] Reeves CR, 2000, *Fitness landscapes and evolutionary algorithms*, pp. 3–20 in Fonlupt C, Hao J, Lutton E, Schoenauer M & Ronald E (Eds), *Artificial evolution*, Springer, Berlin.

[142] Resende MGC & Ribeiro CC, 2003, *Greedy randomized adaptive search procedures*, pp. 219–250 in Glover FW & Kochenberger GA (Eds), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[143] Robbins H & Monro S, 1951, *A stochastic approximation method*, The Annals of Mathematical Statistics, **22(3)**, pp.400–407.

[144] Romeijn HE & Smith RL, 1994, *Simulated annealing for constrained global optimization*, Journal of Global Optimization, **5**, pp.101–126.

[145] Ross SM, 1996, *Stochastic Processes*, Wiley, New York (NY).

[146] Rudolph G, 1994, *Convergence analysis of canonical genetic algorithms*, IEEE Transactions on Neural Networks, **5(1)**, pp.96–101.

[147] Rudolph G, 1996, *Convergence of evolutionary algorithms in general search spaces*, Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, pp. 50–54.

[148] Rudolph G, 1997, *Convergence properties of evolutionary algorithms*, Kovač, Hamburg.

[149] Ruse M, 1996, *Are pictures really necessary? The case of Sewall Wright's 'Adaptive Landscapes'*, pp. 303–337 in Baigrie BS (Ed), *Picturing knowledge: Historical and philosophical problems concerning the use of art in science*, University of Toronto Press, Toronto.

[150] Sánchez AM, Lozano M, García-Martínez C, Molina D & Herrera F, 2008, *Real-parameter crossover operators with multiple descendents: An experimental study*, International Journal of Intelligent Systems, **23(2)**, pp.246–268.

[151] Sareni B & Krahenbuhl L, 1998, *Fitness sharing and niching methods revisited*, IEEE Transactions on Evolutionary Computation, **2(3)**, pp.97–106.

[152] Sarker RA, Mohammadian M & Yao X (Eds), 2002, *Evolutionary optimization*, Kluwer Academic Publishers, Boston (MA).

[153] Schoen F, 2001, *Stochastic global optimization: Two-phase methods*, pp. 301–305 in Floudas CA & Pardalos PM (Eds), *Encyclopedia of optimization*, Kluwer, Dordrecht.

[154] Schrack G & Choit M, 1976, *Optimized relative step size random searches*, Mathematical Programming, **10(2–3)**, pp.230–244.

[155] Schumer MA & Steiglitz K, 1968, *Adaptive step size random search*, IEEE Transactions on Automatic Control, **13(3)**, pp.270–276.

[156] Schwefel H, 1995, *Evolution and optimum seeking*, Wiley, New York.

[157] Senaratna NI, 2005, *Genetic algorithms: The crossover-mutation debate*, BSc Thesis, Univeristy of Colombo, Colombo.

[158] Siarry P & Michalewicz Z, 2008, *Advances in metaheuristics for hard optimization*, Springer, Heidelberg.

[159] Silver N, 2012, *The signal and the noise: Why so many predictions fail — but some don't*, Penguin Group US, New York (NY).

[160] Solis FJ & Wets RJ, 1981, *Minimization by random search techniques*, Mathematics of Operations Research, **6(1)**, pp.19–30.

[161] Spears WM, 1995, *Adapting crossover in evolutionary algorithms*, Proceedings of the 4th Annual Conference on Evolutionary Programming, San Diego (CA), pp. 367–384.

[162] Spears WM, 1992, *Crossover or mutation?*, Proceedings of the Foundations of Genetic Algorithms 2, Vail (CO), pp. 221–237.

[163] Spears WM & Jong KAD, 1998, *Dining with GAs: Operator lunch theorems*, Proceedings of the 5th Workshop on Foundations of Genetic Algorithms, Amsterdam, pp. 85–101.

[164] Stark DR & Spall JC, 2003, *Rate of convergence in evolutionary computation*, Proceedings of the American Control Conference, Denver (CO), pp. 1932–1937.

[165] Stephens CR & Vargas JM, 2000, *Effective fitness as an alternative paradigm for evolutionary computation, I: General formalism*, Genetic Programming and Evolvable Machines, **1(4)**, pp.363–378.

[166] Stützle T, 1998, *Local search algorithms for combinatorial problems — Analysis, improvements, and new applications*, PhD Thesis, Department of Computer Science, Darmstadt University of Technology, Darmstadt.

[167] Sumathi S, Hamsapriya T & Surekha P, 2008, *Evolutionary intelligence*, Springer, London.

[168] Talbi E, 2009, *Metaheuristics, from design to implementation*, John Wiley & Sons, Hoboken (NJ).

[169] Tasoulis DK, Plagianakos VP & Vrahatis MN, 2005, *Clustering in evolutionary algorithms to efficiently compute simultaneously local and global minima*, Proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, pp. 1847–1854.

[170] Torczon V & Trosset MW, 1998, *Using approximations to accelerate engineering design optimization*, (Unpublished) Technical Report, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton (VA).

[171] Ursem RK, 2002, *Diversity-guided evolutionary algorithms*, Proceedings of the Congress on Evolutionary Computation, Honolulu, pp. 1633–1640.

[172] Vose MD, 1995, *Modeling simple genetic algorithms*, Evolutionary Computation, **3(4)**, pp.453–472.

[173] Voudouris C & Tsang E, 1995, *Guided local search*, (Unpublished) Technical Report, Department of Computer Science, University of Essex, Colchester.

[174] Voudouris C & Tsang EPK, 2003, *Guided local search*, pp. 185–218 in Glover F & Kochenberger GA (Eds), *Handbook of metaheuristics*, Kluwer, Dordrecht.

[175] Watson J, 2010, *An introduction to fitness landscape analysis and cost models for local search*, Springer, Boston (MA).

[176] Weise T, 2009, *Global optimization algorithms— Theory and application*, E-book, [Online; Accessed December 5, 2012], URL: `http://www.it-weise.de/projects/book.pdf`.

[177] Wikipedia, 2013, *Lipschitz continuity*, [Online; Accessed July 4, 2013], URL: `http://en.wikipedia.org/wiki/Lipschitz_continuity`.

[178] Wilson C, 2013, *Book review: The signal and the noise*, From Bookforum website [Online; Accessed July 13, 2013], URL: `http://www.bookforum.com/inprint/019_04/10254`.

[179] Wolpert DH & Macready WG, 2005, *Coevolutionary free lunches*, IEEE Transactions on Evolutionary Computation, **9(6)**, pp.721–735.

[180] Wolpert DH & Macready WG, 1997, *No free lunch theorems for optimization*, IEEE Transactions on Evolutionary Computation, **1(1)**, pp.67–82.

[181] Woodward JR, 2003, *GA or GP? That is not the question*, Proceedings of the Congress on Evolutionary Computation, Canberra, pp. 1056–1063.

[182] Wright S, 1930, *Evolution in Mendelian populations*, Genetics, **16**, pp.97–159.

[183] Wright S, 1932, *The roles of mutation, inbreeding, crossbreeding and selection in evolution*, Proceedings of the 6th International Congress of Genetics, Brooklyn (NY), pp. 356–366.

[184] Wu X & Graña M, 2005, *Information processing with evolutionary algorithms: From industrial applications to academic speculations*, Springer-Verlag, London.

[185] Xiao S, Rotaru M & Sykulski JK, 2012, *Exploration versus exploitation using Kriging surrogate modelling in electromagnetic design*, International Journal for Computation and Mathematics in Electrical and Electronic Engineering, **31(5)**, pp.1541–1551.

[186] Yang X, 2012, *Nature-inspired metaheuristic algorithms: Success and new challenges*, Journal of Computer Engineering and Information Technology, **1(1)**, pp.1–3.

[187] Yang X, 2011, *Review of metaheuristics and generalized evolutionary walk algorithm*, International Journal Bio-Inspired Computation, **3**, pp.77–84.

[188] Yen G, Yang F, Hickey T & Goldstein M, 2001, *Coordination of exploration and exploitation in a dynamic environment*, Proceedings of the International Joint Conference on Neural Networks, Baltimore (MD), pp. 1014–1018.

[189] Yuen SY & Chow CK, 2009, *Continuous non-revisiting genetic algorithm*, Proceedings of the IEEE Congress on Evolutionary Computation, Trondheim, pp. 1896–1903.

[190] Yusup N, Zaim AM & Hashim SZM, 2012, *Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007–2011)*, Expert Systems with Applications, **39(10)**, pp.9909–9927.

[191] Zabinsky ZB, 2010, *Random search algorithms*, Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP & Smith JC (Eds), *Wiley encyclopedia of operations research and management science*, John Wiley & Sons, Online, URL: `http://onlinelibrary.wiley.com/doi/10.1002/9780470400531.eorms0704/full`.

[192] Zabinsky ZB, 2003, *Stochastic adaptive search for global optimization*, Kluwer Academic Publishers, Boston (MA).

[193] Zhigljavsky AA, 1992, *Theory of global random search*, Kluwer, Dordrecht.

APPENDIX A

Literature Review

This appendix details the terminology count of the Journal of Heuristics, IEEE Transactions on Evolutionary Computation and Evolutionary Computation for the years 2011 to 2012. The total number of articles in which a term is used (in an appropriate context) over all of the journal articles considered is shown in Table A.1. The final column *Any* refers to the number of papers for which any of the terms are used. Below there are sections detailing the count for each journal.

Journal	*# Papers*	*Exploit-*	*Explor-*	*Intens-*	*Divers-*	*Any*
Journal of Heuristics	64	19	40	18	38	52
IEEE Tr. on Ev. Comp.	100	55	67	8	68	91
Evolutionary Comp.	45	22	23	5	22	32
Total	209	96	130	31	128	175
Percentage	—	46%	62%	15%	61%	84%

TABLE A.1: *Total terminology count.*

A.1 Journal of Heuristics

The terminology count of the Journal of Heuristics is shown in Table A.2. If a term is used at least once in an appropriate context in an article, then there is a "1" in the corresponding cell of the table (if not, then there is a "0").

Paper #	*Exploit-*	*Explor-*	*Intens-*	*Divers-*	*Any*
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	0	0	1
4	0	0	0	0	0
5	1	1	0	0	1
6	0	1	1	1	1
7	1	1	1	1	1
8	0	1	0	1	1
9	1	1	0	1	1
10	0	1	0	1	1
11	0	1	0	1	1
12	1	1	0	1	1

13	0	0	0	1	1
14	0	0	0	0	0
15	1	0	1	1	1
16	0	0	0	0	0
17	0	1	0	0	1
18	0	0	0	0	0
19	0	0	0	0	0
20	0	1	1	1	1
21	0	0	0	0	0
22	0	0	0	0	0
23	1	1	1	1	1
24	0	1	0	0	1
25	0	0	0	1	1
26	0	0	0	1	1
27	1	0	1	1	1
28	0	1	0	1	1
29	0	1	0	1	1
30	1	1	0	1	1
31	1	1	0	1	1
32	0	1	1	1	1
33	0	1	1	1	1
34	0	1	1	0	1
35	0	1	1	1	1
36	0	1	1	1	1
37	0	0	0	0	0
38	0	1	0	0	1
39	1	1	0	1	1
40	0	1	0	0	1
41	1	1	1	1	1
42	1	1	1	1	1
43	0	0	0	0	0
44	0	1	0	0	1
45	0	0	0	1	1
46	0	1	0	0	1
47	0	1	0	1	1
48	0	0	0	1	1
49	0	1	1	1	1
50	1	1	0	0	1
51	0	1	0	1	1
52	0	0	0	1	1
53	0	0	0	1	1
54	1	1	0	0	1
55	1	1	0	0	1
56	1	1	0	1	1
57	0	0	0	1	1
58	1	1	0	1	1
59	0	0	0	1	1
60	0	0	0	0	0
61	0	1	1	1	1
62	0	0	1	1	1

63	1	1	1	0	1
64	0	1	1	0	1
Sum	19	40	18	38	52
Percentage	30%	63%	28%	59%	81%

TABLE A.2: *Terminology count for the Journal of Heuristics.*

A.2 IEEE Transactions on Evolutionary Computation

The terminology count of the IEEE Transactions on Evolutionary Computation is shown in Table A.3. If a term is used at least once in an appropriate context in an article, then there is a "1" in the corresponding cell of the table (if not, then there is a "0").

Paper #	*Exploit-*	*Explor-*	*Intens-*	*Divers-*	*Any*
1	1	1	1	1	1
2	1	1	0	1	1
3	1	1	0	1	1
4	1	1	0	1	1
5	1	1	0	1	1
6	1	1	0	0	1
7	0	0	0	1	1
8	0	0	0	0	0
9	1	1	0	1	1
10	1	1	0	1	1
11	1	0	0	0	1
12	1	0	0	1	1
13	0	1	0	1	1
14	1	0	0	1	1
15	0	1	1	1	1
16	1	1	0	1	1
17	0	1	0	0	1
18	1	1	0	0	1
19	0	1	0	0	1
20	1	1	0	1	1
21	0	0	0	0	0
22	0	1	0	1	1
23	0	1	0	0	1
24	0	0	0	1	1
25	1	1	0	1	1
26	0	1	0	1	1
27	1	1	0	1	1
28	1	1	0	1	1
29	0	0	0	1	1
30	1	0	0	1	1
31	0	0	0	0	0
32	0	0	0	1	1
33	1	0	0	1	1
34	0	1	0	1	1

35	1	1	1	1	1
36	1	1	1	1	1
37	0	0	0	0	0
38	0	0	0	0	0
39	1	1	0	1	1
40	1	1	1	1	1
41	1	1	0	0	1
42	1	1	0	1	1
43	0	1	0	0	1
44	0	0	0	1	1
45	1	1	1	1	1
46	1	0	0	1	1
47	0	0	0	1	1
48	0	0	0	0	0
49	0	0	0	1	1
50	0	0	0	1	1
51	0	1	0	0	1
52	0	0	0	0	0
53	0	0	0	1	1
54	1	0	0	0	1
55	1	1	0	1	1
56	1	1	0	1	1
57	1	0	0	1	1
58	1	1	0	1	1
59	1	1	0	1	1
60	1	1	0	0	1
61	1	1	0	1	1
62	1	1	0	1	1
63	0	1	0	1	1
64	1	1	0	1	1
65	0	1	0	0	1
66	0	1	1	1	1
67	0	0	0	0	0
68	1	1	1	1	1
69	1	0	0	0	1
70	0	0	0	1	1
71	1	1	0	1	1
72	0	1	0	1	1
73	0	0	0	1	1
74	1	0	0	1	1
75	0	1	0	1	1
76	0	1	0	0	1
77	0	1	0	0	1
78	1	1	0	0	1
79	1	1	0	1	1
80	1	1	0	1	1
81	0	1	0	0	1
82	1	1	0	1	1
83	1	0	0	1	1
84	0	0	0	1	1

85	1	1	0	1	1
86	1	1	0	1	1
87	1	1	0	1	1
88	0	0	0	1	1
89	0	1	0	0	1
90	1	1	0	0	1
91	0	1	0	0	1
92	1	1	0	1	1
93	0	0	0	0	0
94	1	1	0	1	1
95	0	1	0	1	1
96	0	1	0	1	1
97	1	1	0	0	1
98	0	1	0	0	1
99	1	1	0	1	1
100	1	1	0	0	1
Sum	55	67	8	68	91
Percentage	55%	67%	8%	68%	91%

TABLE A.3: *Terminology count for the IEEE Transactions on Evolutionary Computation.*

A.3 Evolutionary Computation

The terminology count of the Evolutionary Computation is shown in Table A.4. If a term is used at least once in an appropriate context in an article, then there is a "1" in the corresponding cell of the table (if not, then there is a "0").

Paper #	*Exploit-*	*Explor-*	*Intens-*	*Divers-*	*Any*
1	0	0	0	0	0
2	1	1	0	0	1
3	0	0	0	1	1
4	1	0	0	0	0
5	0	0	0	1	1
6	1	1	0	1	1
7	0	0	0	0	0
8	0	1	0	0	1
9	1	1	0	1	1
10	1	1	0	1	1
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	1	0	0	1	1
15	1	1	0	1	1
16	0	0	1	1	1
17	0	0	0	0	0
18	1	1	0	1	1
19	0	1	1	1	1
20	0	0	0	0	0

21	1	0	0	1	1
22	1	1	0	0	1
23	0	0	0	0	0
24	0	0	0	1	1
25	0	1	0	1	1
26	0	0	0	0	0
27	1	1	0	1	1
28	1	1	0	1	1
29	1	0	1	1	1
30	1	1	0	0	1
31	0	0	0	0	0
32	1	1	1	1	1
33	0	1	0	0	1
34	1	0	0	0	1
35	0	1	0	0	1
36	1	1	0	1	1
37	0	0	0	0	0
38	0	0	0	1	1
39	1	1	0	0	1
40	1	1	1	1	1
41	0	1	0	1	1
42	1	1	0	0	1
43	1	1	0	1	1
44	1	1	0	0	1
45	0	0	0	0	0
Sum	22	23	5	22	32
Percentage	49%	51%	11%	49%	71%

TABLE A.4: *Terminology count for Evolutionary Computation.*

References from the Journal of Heuristics

1. Ouzineb, M., Nourelfath, M. and Gendreau, M. (2011) 'A heuristic method for non-homogeneous redundancy optimization of series-parallel multi-state systems', Vol. 17, No. 1.
2. Gouveia, L., Paias, A. and Sharma, D. (2011) 'Restricted dynamic programming based neighborhoods for the hop-constrained minimum spanning tree problem', Vol. 17, No. 1.
3. Bansal, R. and Srivastava, K. (2011) 'Memetic algorithm for the antibandwidth maximization problem', Vol. 17, No. 1.
4. Deĭneko, V.G., Shabtay, D. and Steiner, G. (2011) 'On the asymptotic behavior of subtour-patching heuristics in solving the TSP on permuted Monge matrices', Vol. 17, No. 1.
5. Lü, Z., Hao, J.-K. and Glover, F. (2011) 'Neighborhood analysis: a case study on curriculum-based course timetabling', Vol. 17, No. 2.
6. Deng, Y. and Bard, J.F. (2011) 'A reactive GRASP with path relinking for capacitated clustering', Vol. 17, No. 2.
7. Aloise, D. and Ribeiro, C.C. (2011) 'Adaptive memory in multistart heuristics for multicommodity network design', Vol. 17, No. 2.
8. Pullan, W., Mascia, F. and Brunato, M. (2011) 'Cooperating local search for the maximum clique problem', Vol. 17, No. 2.

9. Karapetyan, D. and Gutin, G. (2011) 'Local search heuristics for the multidimensional assignment problem', Vol. 17, No. 3.
10. Hamadi, Y. and Ringwelski, G. (2011) 'Boosting distributed constraint satisfaction', Vol. 17, No. 3.
11. Paletta, G. and Vocaturo, F. (2011) 'A composite algorithm for multiprocessor scheduling', Vol. 17, No. 3.
12. Zhang, G. (2011) 'Quantum-inspired evolutionary algorithms: a survey and empirical study', Vol. 17, No. 3.
13. Van Laarhoven, J.W. and Ohlmann, J.W. (2011) 'A randomized Delaunay triangulation heuristic for the Euclidean Steiner tree problem in Rd', Vol. 17, No. 4.
14. Gilli, M. and Schumann, E. (2011) 'Optimal enough?', Vol. 17, No. 4.
15. Riise, A. and Burke, E.K. (2011) 'Local search for the surgery admission planning problem', Vol. 17, No. 4.
16. Dotu, I., Patricio, M.A., Berlanga, A., García, J. and Molina, J.M. (2011) 'Boosting video tracking performance by means of Tabu Search in intelligent visual surveillance systems', Vol. 17, No. 4.
17. De Leone, R., Festa, P. and Marchitto, E. (2011) 'A Bus Driver Scheduling Problem: a new mathematical model and a GRASP approximate solution', Vol. 17, No. 4.
18. Gonçalves, J.F., Resende, M.G.C. and Mendes, J.J.M. (2011) 'A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem', Vol. 17, No. 5.
19. Gonçalves, J.F. and Resende, M.G.C. (2011) 'Biased random-key genetic algorithms for combinatorial optimization', Vol. 17, No. 5.
20. Mateus, G.R., Resende, M.G.C. and Silva, R.M.A. (2011) 'GRASP with path-relinking for the generalized quadratic assignment problem', Vol. 17, No. 5.
21. Bengoetxea, E., Larrañaga, P., Bielza, C. and Fernández del Pozo, J.A. (2011) 'Optimal row and column ordering to improve table interpretation using estimation of distribution algorithms', Vol. 17, No. 5.
22. Ansótegui, C., Béjar, R., Fernández, C., Gomes, C. and Mateu, C. (2011) 'Generating highly balanced sudoku problems as hard problems', Vol. 17, No. 5.
23. Toril, M., Wille, V., Molina-Fernández, I. and Walshaw, C. (2011) 'An adaptive multi-start graph partitioning algorithm for structuring cellular networks', Vol. 17, No. 5.
24. Brueggemann, T. and Hurink, J.L. (2011) 'Matching based very large-scale neighborhoods for parallel machine scheduling', Vol. 17, No. 5.
25. Fadlaoui, K. and Galinier, P. (2011) 'A tabu search algorithm for the covering design problem', Vol. 17, No. 6.
26. Lin, C., Qing, A. and Feng, Q. (2011) 'A comparative study of crossover in differential evolution', Vol. 17, No. 6.
27. Van Peteghem, V. and Vanhoucke, M. (2011) 'Using resource scarceness characteristics to solve the multi-mode resource-constrained project scheduling problem', Vol. 17, No. 6.
28. Labadie, N., Melechovský, J. and Calvo, R.W. (2011) 'Hybridized evolutionary local search algorithm for the team orienteering problem with time windows', Vol. 17, No. 6.
29. Abdullah, S., Turabieh, H., McCollum, B. and McMullan, P. (2012) 'A hybrid metaheuristic approach to the university course timetabling problem', Vol. 18, No. 1.
30. Czogalla, J. and Fink, A. (2012) 'Fitness landscape analysis for the no-wait flow-shop scheduling problem', Vol. 18, No. 1.
31. Mateo, P. M. and Alberto, I. (2012) 'A mutation operator based on a Pareto ranking for multi-objective evolutionary algorithms', Vol. 18, No. 1.

32. Sun, M. (2012) 'A tabu search heuristic procedure for the capacitated facility location problem', Vol. 18, No. 1.
33. Chiarandini, M., Di Gaspero, L., Gualandi, S. and Schaerf, A. (2012) 'The balanced academic curriculum problem revisited', Vol. 18, No. 1.
34. Soto, M., Rossi, A. and Sevaux, M. (2012) 'A mathematical model and a metaheuristic approach for a memory allocation problem', Vol. 18, No. 1.
35. Santamara, J., Cordón, O., Damas, S., Martí, R. and Palma, R. J. (2012) 'GRASP and path relinking hybridizations for the point matching-based image registration problem', Vol. 18, No. 1.
36. Ribeiro, C.C. and Resende, M.G.C. (2012) 'Path-relinking intensification methods for stochastic local search algorithms', Vol. 18, No. 2.
37. Ben-Moshe, B. (2012) 'Geometric heuristics for rural radio maps approximation', Vol. 18, No. 2.
38. Mousavi, S.R. Babaie, M. and Montazerian, M. (2012) 'An improved heuristic for the far from most strings problem', Vol. 18, No. 2.
39. Basseur, M., Liefooghe, A., Le, K. and Burke, E.K. (2012) 'The efficiency of indicator-based local search for multi-objective combinatorial optimisation problems', Vol. 18, No. 2.
40. Rainwater, C., Geunes, J. and Romeijn, H.E. (2012) 'A facility neighborhood search heuristic for capacitated facility location with single-source constraints and flexible demand', Vol. 18, No. 2.
41. Liefooghe, A., Humeau, J., Mesmoudi, S., Jourdan, L. and Talbi, E.-G. (2012) 'On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems', Vol. 18, No. 2.
42. Kiziltan, Z., Lodi, A., Milano, M. and Parisini, F. (2012) 'Bounding, filtering and diversification in CP-based local branching', Vol. 18, No. 3.
43. Minis, I., Mamasis, K. and Zeimpekis, V. (2012) 'Real-time management of vehicle breakdowns in urban freight distribution', Vol. 18, No. 3.
44. Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W. and Vanden Berghe, G. (2012) 'One hyper-heuristic approach to two timetabling problems in health care', Vol. 18, No. 3.
45. Kimms, A. and Maassen, K.-C. (2012) 'Cell-transmission-based evacuation planning with rescue teams', Vol. 18, No. 3.
46. LaRusic, J., Punnen, A.P. and Aubanel, E. (2012) 'Experimental analysis of heuristics for the bottleneck traveling salesman problem', Vol. 18, No. 3.
47. Moreira, M.C.O., Ritt, M., Costa, A.M. and Chaves, A.A. (2012) 'Simple heuristics for the assembly line worker assignment and balancing problem', Vol. 18, No. 3.
48. Andrade, D.V., Resende, M.G.C. and Werneck, R.F. (2012) 'Fast local search for the maximum independent set problem', Vol. 18, No. 4.
49. Davidović, T., Šelmić, M.,Teodorović, D. and Ramljak, D. (2012) 'Bee colony optimization for scheduling independent tasks to identical processors', Vol. 18, No. 4.
50. Liang, Y.-C., Lee, Z.-H. and Chen, Y.-S. (2012) 'A novel ant colony optimization approach for on-line scheduling and due date determination', Vol. 18, No. 4.
51. Kasperski, A., Makuchowski, M. and Zieliński, P. (2012) 'A tabu search algorithm for the minmax regret minimum spanning tree problem with interval data', Vol. 18, No. 4.
52. Mendes, L., Godinho, P. and Dias, J. (2012) 'A Forex trading system based on a genetic algorithm', Vol. 18, No. 4.
53. Santos, A.C., Duhamel, C., Belisário, L.S. and Guedes, L.M. (2012) 'Strategies for designing energy-efficient clusters-based WSN topologies', Vol. 18, No. 4.
54. Chen, J., Zhu, W. and Peng, Z. (2012) 'A heuristic algorithm for the strip packing problem', Vol. 18, No. 4.

55. Vázquez-Rodríguez, J.A. and Petrovic, S. (2012) 'Calibrating continuous multi-objective heuristics using mixture experiments', Vol. 18, No. 5.

56. Drugan, M.M. and Thierens, D. (2012) 'Stochastic Pareto local search: Pareto neighbourhood exploration and perturbation strategies', Vol. 18, No. 5.

57. Dang, D.-C. and Moukrim, A. (2012) 'Subgraph extraction and metaheuristics for the maximum clique problem', Vol. 18, No. 5.

58. Larrañaga, P., Karshenas, H., Bielz, C. and Santana, R. (2012) 'A review on probabilistic graphical models in evolutionary computation', Vol. 18, No. 5.

59. Goel, V., Furman, K.C., Song, J.-H. and El-Bakry, A.S. (2012) 'Large neighborhood search for LNG inventory routing', Vol. 18, No. 6.

60. DellAmico, M., Iori, M., Martello, S. and Monaci, M. (2012) 'A note on exact and heuristic algorithms for the identical parallel machine scheduling problem', Vol. 18, No. 6.

61. Rodríguez-Martín, I. and Salazar-González, J. J. (2012) 'A hybrid heuristic approach for the multi-commodity one-to-one pickup-and-delivery traveling salesman problem', Vol. 18, No. 6.

62. Voß, S. and Fink, A. (2012) 'A hybridized tabu search approach for the minimum weight vertex cover problem', Vol. 18, No. 6.

63. Guastaroba, G. and Speranza, M.G. (2012) 'Kernel search for the capacitated facility location problem', Vol. 18, No. 6.

64. Lozano, M., Duarte, A., Gortázar, F. and Martí, R. (2012) 'Variable neighborhood search with ejection chains for the antibandwidth problem', Vol. 18, No. 6.

References from the IEEE Transactions on Evolutionary Computation

1. Das, S. and Suganthan, P.N. (2011) 'Differential Evolution: A Survey of the State-of-the-Art', Vol. 15, No. 1.

2. Mininno, E., Neri, F., Cupertino, F. and Naso, D. (2011) 'Compact Differential Evolution', Vol. 15, No. 1.

3. Epitropakis, M.G., Tasoulis, D.K., Pavlidis, N.G., Plagianakos, V.P. and Vrahatis, M.N. (2011) 'Enhancing Differential Evolution Utilizing Proximity-Based Mutation Operators', Vol. 15, No. 1.

4. Wang, Y., Cai, Z. and Zhang, Q. (2011) 'Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters', Vol. 15, No. 1.

5. Dorronsoro, B. and Bouvry, P. (2011) 'Improving Classical and Decentralized Differential Evolution with New Mutation Operator and Population Topologies', Vol. 15, No. 1.

6. Das, S., Suganthan, P.N., Coello Coello, C.A. (2011) 'Guest Editorial: Special Issue on Differential Evolution', Vol. 15, No. 1.

7. Wang, L. and Li, L.-P. (2011) 'Fixed-Structure H_∞ Controller Synthesis Based on Differential Evolution with Level Comparison', Vol. 15, No. 1.

8. Orlov, M. and Sipper, M. (2011) 'Flight of the FINCH through the Java Wilderness', Vol. 15, No. 2.

9. Adra, S.F. and Fleming, P.J. (2011) 'Diversity Management in Evolutionary Many-Objective Optimization', Vol. 15, No. 2.

10. Tometzki, T. and Engell, S. (2011) 'Systematic Initialization Techniques for Hybrid Evolutionary Algorithms for Solving Two-Stage Stochastic Mixed-Integer Programs', Vol. 15, No. 2.

11. Aydt, H., Turner, S.J., Cai, W., Low, M.Y.H., Ong, Y.-S. and Ayani, R. (2011) 'Toward an Evolutionary Computing Modeling Language', Vol. 15, No. 2.

12. Cartlidge, J. and Ait-Boudaoud, D. (2011) 'Autonomous Virulence Adaptation Improves Coevolutionary Optimization', Vol. 15, No. 2.

13. Someya, H. (2011) 'Theoretical Analysis of Phenotypic Diversity in Real-Valued Evolutionary Algorithms with More-Than-One-Element Replacement', Vol. 15, No. 2.

14. Mei, Y., Tang, K. and Yao, X. (2011) 'Decomposition-Based Memetic Algorithm for Multiobjective Capacitated Arc Routing Problem', Vol. 15, No. 2.

15. Vasile, M., Minisci, E. and Locatelli, M. (2011) 'An Inflationary Differential Evolution Algorithm for Space Trajectory Optimization', Vol. 15, No. 2.

16. Lee, D.S., Gonzalez, L.F., Périaux, J. and Srinivas, K. (2011) 'Efficient Hybrid-Game Strategies Coupled to Evolutionary Algorithms for Robust Multidisciplinary Design Optimization in Aerospace Engineering', Vol. 15, No. 2.

17. Fernandez-Martinez, J.L. and Garcia-Gonzalo, E. (2011) 'Stochastic Stability Analysis of the Linear Continuous and Discrete PSO Models', Vol. 15, No. 3.

18. Bush, B.J. and Sayama, H. (2011) 'Hyperinteractive Evolutionary Computation', Vol. 15, No. 3.

19. Hoang, T.-H., McKay, R.I., Essam, D. and Nguyen Xuan Hoai (2011) 'On Synergistic Interactions Between Evolution, Development and Layered Learning', Vol. 15, No. 3.

20. Clune, J., Stanley, K.O., Pennock, R.T. and Ofria, C. (2011) 'On the Performance of Indirect Encoding Across the Continuum of Regularity', Vol. 15, No. 3.

21. Kuyucu, T., Trefzer, M.A., Miller, J.F. and Tyrrell, A.M. (2011) 'An Investigation of the Importance of Mechanisms and Parameters in a Multicellular Developmental System', Vol. 15, No. 3.

22. Devert, A., Bredeche, N. and Schoenauer, M. (2011) 'Robustness and the Halting Problem for Multicellular Artificial Ontogeny', Vol. 15, No. 3.

23. Valsalam, V.K. and Miikkulainen, R. (2011) 'Evolving Symmetry for Modular System Design', Vol. 15, No. 3.

24. Schutze, O., Lara, A. and Coello, C.A.C. (2011) 'On the Influence of the Number of Objectives on the Hardness of a Multiobjective Optimization Problem', Vol. 15, No. 4.

25. Araujo, L. and Merelo, J.J. (2011) 'Diversity Through Multiculturality: Assessing Migrant Choice Policies in an Island Model', Vol. 15, No. 4.

26. Olson, C.C., Nichols, J.M., Todd, M.D., Michalowicz, J.V. and Bucholtz, F. (2011) 'Coupling Evolutionary Algorithms With Vol. 15, Nonlinear Dynamical Systems: An Efficient Tool for Excitation Design and Optimization', Vol. 15, No. 4.

27. White, D.R., Arcuri, A. and Clark, J.A. (2011) 'Evolutionary Improvement of Programs', Vol. 15, No. 4.

28. McConaghy, T., Palmers, P., Steyaert, M. and Gielen, G.G.E. (2011) 'Trustworthy Genetic Programming-Based Synthesis of Analog Circuit Topologies Using Hierarchical Domain-Specific Building Blocks', Vol. 15, No. 4.

29. Singh, H.K., Isaacs, A. and Ray, T. (2011) 'A Pareto Corner Search Evolutionary Algorithm and Dimensionality Reduction in Many-Objective Optimization Problems', Vol. 15, No. 4.

30. Bongard, J.C. (2011) 'Innocent Until Proven Guilty: Reducing Robot Shaping from Polynomial to Linear Time', Vol. 15, No. 4.

31. Gibney, A.M., Klepal, M. and Pesch, D. (2011) 'Agent-Based Optimization for Large Scale WLAN Design', Vol. 15, No. 4.

32. Gang Yu, Tianyou Chai and Xiaochuan Luo (2011) 'Multiobjective Production Planning Optimization Using Hybrid Evolutionary Algorithms for Mineral Processing', Vol. 15, No. 4.

33. Kwasnicka, H. and Przewozniczek, M. (2011) 'Multi Population Pattern Searching Algorithm: A New Evolutionary Method Based on the Idea of Messy Genetic Algorithm', Vol. 15, No. 5.

34. Benlic, U. and Jin-Kao Hao (2011) 'A Multilevel Memetic Approach for Improving Graph k-Partitions', Vol. 15, No. 5.

35. Jih-Yiing Lin and Ying-ping Chen (2011) 'Analysis on the Collaboration Between Global Search and Local Search in Memetic Computation', Vol. 15, No. 5.

36. Zexuan Zhu, Jiarui Zhou, Zhen Ji and Yu-hui Shi (2011) 'DNA Sequence Compression Using Adaptive Particle Swarm Optimization-Based Memetic Algorithm', Vol. 15, No. 5.

37. Urselmann, M., Barkmann, S., Sand, G. and Engell, S. (2011) 'A Memetic Algorithm for Global Optimization in Chemical Process Synthesis Problems', Vol. 15, No. 5.

38. Kyungrock Paik (2011) 'Optimization Approach for 4-D Natural Landscape Evolution', Vol. 15, No. 5.

39. McGinley, B., Maher, J., O'Riordan, C. and Morgan, F. (2011) 'Maintaining Healthy Population Diversity Using Adaptive Crossover, Mutation, and Selection', Vol. 15, No. 5.

40. Xianshun Chen, Yew-Soon Ong, Meng-Hiot Lim and Kay Chen Tan (2011) 'A Multi-Facet Survey on Memetic Computation', Vol. 15, No. 5.

41. Verel, S., Ochoa, G. and Tomassini, M. (2011) 'Local Optima Networks of NK Landscapes With Neutrality', Vol. 15, No. 6.

42. Zhiwen Yu, Hau-San Wong, Dingwen Wang and Ming Wei (2011) 'Neighborhood Knowledge-Based Evolutionary Algorithm for Multiobjective Optimization Problems', Vol. 15, No. 6.

43. Tsung-Ying Sun, Chan-Cheng Liu, Shang-Jeng Tsai, Sheng-Ta Hsieh and Kan-Yuan Li (2011) 'Cluster Guide Particle Swarm Optimization (CGPSO) for Underdetermined Blind Source Separation with Advanced Conditions', Vol. 15, No. 6.

44. Zhi-hui Zhan, Jun Zhang, Yun Li and Yu-hui Shi (2011) 'Orthogonal Learning Particle Swarm Optimization', Vol. 15, No. 6.

45. Brunato, M. and Battiti, R. (2011) 'R-evo: A Reactive Evolutionary Algorithm for the Maximum Clique Problem', Vol. 15, No. 6.

46. Chi Kin Chow and Shiu Yin Yuen (2011) 'An Evolutionary Algorithm That Makes Decision Based on the Entire Previous Search History', Vol. 15, No. 6.

47. Carrano, E.G., Wanner, E.F. and Takahashi, R.H.C. (2011) 'A Multicriteria Statistical Based Comparison Methodology for Evaluating Evolutionary Algorithms ', Vol. 15, No. 6.

48. Bui, L T., Abbass, H.A., Barlow, M. and Bender, A. (2012) 'Robustness Against the Decision-Maker's Attitude to Risk in Problems With Conflicting Objectives', Vol. 16, No. 1.

49. Kim, J.-H., Han, J.-H., Kim, Y.-H., Choi, S.-H., and Kim, E.-S. (2012) 'Preference-Based Solution Selection Algorithm for Evolutionary Multiobjective Optimization', Vol. 16, No. 1.

50. While, L., Bradstreet, L. and Barone, L. (2012) 'A Fast Way of Calculating Exact Hypervolumes', Vol. 16, No. 1.

51. Ramirez, R., Maestre, E. and Serra, X. (2012) 'A Rule-Based Evolutionary Approach to Music Performance Modeling', Vol. 16, No. 1.

52. Steitz, W. and Rothlauf, F. (2012) 'Edge Orientation and the Design of Problem-Specific Crossover Operators for the OCST Problem', Vol. 16, No. 1.

53. Chong, S.Y., Tiňo, P., Ku, D.C. and Yao, X. (2012) 'Improving Generalization Performance in Co-Evolutionary Learning', Vol. 16, No. 1.

54. Vural, R.A., Yildirim, T., Kadioglu, T. and Basargan, A. (2012) 'Performance Evaluation of Evolutionary Algorithms for Optimal Filter Design', Vol. 16, No. 1.

55. Bosman, P.A.N. (2012) 'On Gradients and Hybrid Evolutionary Algorithms for Real-Valued Multiobjective Optimization', Vol. 16, No. 1.

56. Shang, R., Jiao, L., Liu, F. and Ma, W. (2012) 'A Novel Immune Clonal Algorithm for MO Problems', Vol. 16, No. 1.

57. Wang, Y. and Cai, Z. (2012) 'Combining Multiobjective Optimization with Differential Evolution to Solve Constrained Optimization Problems', Vol. 16, No. 1.

58. Echegoyen, C., Mendiburu, A., Santana, R. and Lozano, J.A. (2012) 'Toward Understanding EDAs Based on Bayesian Networks Through a Quantitative Analysis', Vol. 16, No. 2.

59. Chow, C.K. and Yuen, S.Y. (2012) 'A Multiobjective Evolutionary Algorithm That Diversifies Population by Its Density', Vol. 16, No. 2.

60. Bui, T.N. Deng, X. and Zrncic, C.M. (2012) 'An Improved Ant-Based Algorithm for the Degree-Constrained Minimum Spanning Tree Problem', Vol. 16, No. 2.

61. Lehre, P.K. and Yao, X. (2012) 'On the Impact of Mutation-Selection Balance on the Runtime of Evolutionary Algorithms', Vol. 16, No. 2.

62. Li, X. and Yao, X. (2012) 'Cooperatively Coevolving Particle Swarms for Large Scale Optimization', Vol. 16, No. 2.

63. Weise, T. and Tang, K. (2012) 'Evolving Distributed Algorithms with Genetic Programming', Vol. 16, No. 2.

64. Bui, L.T. Michalewicz, Z.,Parkinson, E. and Abello, M.B. (2012) 'Adaptation in Dynamic Environments: A Case Study in Mission Planning', Vol. 16, No. 2.

65. Poli, R. and Galván-López, E. (2012) 'The Effects of Constant and Bit-Wise Neutrality on Problem Hardness, Fitness Distance Correlation and Phenotypic Mutation Rates', Vol. 16, No. 2.

66. Blackwell, T. (2012) 'A Study of Collapse in Bare Bones Particle Swarm Optimization', Vol. 16, No. 3.

67. Burke, E.K., Hyde, M.R. and Kendall, G. (2012) 'Grammatical Evolution of Local Search Heuristics', Vol. 16, No. 3.

68. Lam, A.Y.S., Li, V.O.K. and Yu, J.J.Q. (2012) 'Real-Coded Chemical Reaction Optimization', Vol. 16, No. 3.

69. Lochtefeld, D.F. and Ciarallo, F.W. (2012) 'Multiobjectivization via Helper-Objectives with the Tunable Objectives Problem', Vol. 16, No. 3.

70. Smith, J.E., Clark, A.R., Staggemeier, A.T. and Serpell, M.C. (2012) 'A Genetic Approach to Statistical Disclosure Control', Vol. 16, No. 3.

71. Dupuis, J.-F., Fan, Z. and Goodman, E.D. (2012) 'Evolutionary Design of Both Topologies and Parameters of a Hybrid Dynamical System', Vol. 16, No. 3.

72. Pizzuti, C. (2012) 'A Multiobjective Genetic Algorithm to Find Communities in Complex Networks', Vol. 16, No. 3.

73. Prügel-Bennett, A. and Tayarani-Najaran, M.-H. (2012) 'Maximum Satisfiability: Anatomy of the Fitness Landscape for a Hard Combinatorial Optimization Problem', Vol. 16, No. 3.

74. Zhao, S.-H., Suganthan, P.N. and Zhang, Q. (2012) 'Decomposition-Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighborhood Sizes', Vol. 16, No. 3.

75. de Mello Honório, L., da Silva, A.M.L. and Barbosa, D.A. (2012) 'A Cluster and Gradient-Based Artificial Immune System Applied in Optimization Scenarios', Vol. 16, No. 3.

76. Karaman, S., Shima, T. and Frazzoli, E. (2012) 'A Process Algebra Genetic Algorithm', Vol. 16, No. 4.

77. Beyer, H.-G. and Finck, S. (2012) 'On the Design of Constraint Covariance Matrix Self-Adaptation Evolution Strategies Including a Cardinality Constraint', Vol. 16, No. 4.

78. Chiong, R. and Kirley, M. (2012) 'Effects of Iterated Interactions in Multiplayer Spatial Evolutionary Games', Vol. 16, No. 4.

79. Kleeman, M.P., Seibert, B.A., Lamont, G.B., Hopkinson, K.M. and Graham, S.R. (2012) 'Solving Multicommodity Capacitated Network Design Problems Using Multiobjective Evolutionary Algorithms', Vol. 16, No. 4.
80. Kohl, N. and Miikkulainen, R. (2012) 'An Integrated Neuroevolutionary Approach to Reactive Control and High-Level Strategy', Vol. 16, No. 4.
81. Kowaliw, T., Dorin, A. and McCormack, J. (2012) 'Promoting Creative Design in Interactive Evolutionary Computation', Vol. 16, No. 4.
82. Li, C. and Yang, S. (2012) 'A General Framework of Multipopulation Methods with Clustering in Undetectable Dynamic Environments', Vol. 16, No. 4.
83. Schütze, O., Esquivel, X., Lara, A. and Coello Coello, C.A. (2012) 'Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization', Vol. 16, No. 4.
84. Arabas, J. (2012) 'Approximating the Genetic Diversity of Populations in the Quasi-Equilibrium State', Vol. 16, No. 5.
85. Arias-Montanõ, A., Coello Coello, C.A. and Mezura-Montes, E. (2012) 'Multiobjective Evolutionary Algorithms in Aeronautical and Aerospace Engineering', Vol. 16, No. 5.
86. Chan, T.-M., Leung, K.-S. and Lee, K.-H. (2012) 'Memetic Algorithms for De Novo Motif Discovery', Vol. 16, No. 5.
87. Corriveau, G., Guilbault, R., Tahan, A. and Sabourin, R. (2012) 'Review and Study of Genotypic Diversity Measures for Real-Coded Representations', Vol. 16, No. 5.
88. Joó, A., Ekárt, A. and Neirotti, J.P. (2012) 'Genetic Algorithms for Discovery of Matrix Multiplication Methods', Vol. 16, No. 5.
89. Howard, G., Gale, E., Bull, L., de Lacy Costello, B. and Adamatzky, A. (2012) 'Evolution of Plastic Learning in Spiking Networks via Memristive Connections', Vol. 16, No. 5.
90. Naznin, F., Sarker, R. and Essam, D. (2012) 'Progressive Alignment Method Using Genetic Algorithm for Multiple Sequence Alignment', Vol. 16, No. 5.
91. Neshatian, K., Zhang, M. and Andreae, P. (2012) 'A Filter Approach to Multiple Feature Construction for Symbolic Learning Classifiers Using Genetic Programming', Vol. 16, No. 5.
92. Qu, B.Y., Suganthan, P.N. and Liang, J.J. (2012) 'Differential Evolution with Neighborhood Mutation for Multimodal Optimization', Vol. 16, No. 5.
93. Gallagher, J.C., Doman, D.B. and Oppenheimer, M.W. (2012) 'The Technology of the Gaps: An Evolvable Hardware Synthesized Oscillator for the Control of a Flapping-Wing Micro Air Vehicle', Vol. 16, No. 6.
94. Nguyen, T.T. and Yao, X. (2012) 'Continuous Dynamic Constrained Optimization — The Challenges', Vol. 16, No. 6.
95. Rodriguez, F.J., García-Martínez, C. and Lozano, M. (2012) 'Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy, Comparison, and Synergy Test', Vol. 16, No. 6.
96. Gong, Y.-J., Zhang, J., Chung, S.-H., Chen, W.-N., Zhan, Z.-H., Li, Y. and Shi, Y.-H. (2012) 'An Efficient Resource Allocation Scheme Using Particle Swarm Optimization', Vol. 16, No. 6.
97. Bull, L. (2012) 'Evolving Boolean Networks on Tunable Fitness Landscapes', Vol. 16, No. 6.
98. Delbem, A.C.B., de Lima, T.W. and Telles, G.P. (2012) 'Efficient Forest Data Structure for Evolutionary Algorithms Applied to Network Design', Vol. 16, No. 6.
99. Dudek, G. (2012) 'An Artificial Immune System for Classification with Local Feature Selection', Vol. 16, No. 6.
100. López-Ibáñez, M. and Stützle, T. (2012) 'The Automatic Design of Multiobjective Ant Colony Optimization Algorithms', Vol.16, No. 6.

References from Evolutionary Computation

1. Hornby, G.S., Lohn, J.D. and Linden, D.S. (2011) 'Computer-Automated Evolution of an X-Band Antenna for NASAs Space Technology 5 Mission', Vol. 19, No. 1.
2. Soule, T. (2011) 'Evolutionary Dynamics of Tag Mediated Cooperation with Multilevel Selection', Vol. 19, No. 1.
3. Bader, J. and Zitzler, E. (2011) 'HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization', Vol. 19, No. 1.
4. Hu, X.-B. and Di Paolo, E.A. (2011) 'A Ripple-Spreading Genetic Algorithm for the Aircraft Sequencing Problem', Vol. 19, No. 1.
5. McIntyre, A.R. and Heywood, M.I. (2011) 'Classification as Clustering: A Pareto Cooperative-Competitive GP Approach', Vol. 19, No. 1.
6. Yeguas, E., Joan-Arinyo, R. and Luzón, M.V. (2011) 'Modeling the Performance of Evolutionary Algorithms on the Root Identification Problem: A Case Study with PBIL and CHC Algorithms', Vol. 19, No. 1.
7. Simon, D. (2011) 'A Probabilistic Analysis of a Simplified Biogeography-Based Optimization Algorithm', Vol. 19, No. 2.
8. Rastegar, R. (2011) 'On the Optimal Convergence Probability of Univariate Estimation of Distribution Algorithms', Vol. 19, No. 2.
9. Wang, Y. and Cai, Z. (2011) 'Constrained Evolutionary Optimization by Means of $(\mu + \lambda)$-Differential Evolution and Improved Adaptive Trade-Off Model', Vol. 19, No. 2.
10. Lehman, J. and Stanley, K.O. (2011) 'Abandoning Objectives: Evolution Through the Search for Novelty Alone', Vol. 19, No. 2.
11. Sung, C.W. and Yuen, S.Y. (2011) 'Analysis of (1+1) Evolutionary Algorithm and Randomized Local Search with Memory', Vol. 19, No. 2.
12. Storch, T. (2011) 'Finding Mount Everest and Handling Voids', Vol. 19, No. 2.
13. Khan, G.M., Miller, J.F. and Halliday, D.M. (2011) 'Evolution of Cartesian Genetic Programs for Development of Learning Neural Architecture', Vol. 19, No. 3.
14. Karapetyan, D. and Gutin, G. (2011) 'A New Approach to Population Sizing for Memetic Algorithms: A Case Study for the Multidimensional Assignment Problem', Vol. 19, No. 3.
15. Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T. and Stanley, K. O. (2011) 'Picbreeder: A Case Study in Collaborative Evolutionary Exploration of Design Space', Vol. 19, No. 3.
16. Li, J., Parkes, A.J. and Burke, E.K. (2011) 'Evolutionary Squeaky Wheel Optimization: A New Framework for Analysis', Vol. 19, No. 3.
17. López-Ibáñez, M., Prasad, T.D. and Paechter, B. (2011) 'Representations and Evolutionary Operators for the Scheduling of Pump Operations in Water Distribution Networks', Vol. 19, No. 3.
18. Coelho, R.F. and Bouillard, P. (2011) 'Multi-Objective Reliability-Based Optimization with Stochastic Metamodels', Vol. 19, No. 4.
19. Li, H. and Landa-Silva, D. (2011) 'An Adaptive Evolutionary Multi-Objective Approach Based on Simulated Annealing', Vol. 19, No. 4.
20. Chicano, F., Whitley, L.D. and Alba, E. (2011) 'A Methodology to Find the Elementary Landscape Decomposition of Combinatorial Optimization Problems', Vol. 19, No. 4.
21. Lewis, R. and Pullin, E. (2011) 'Revisiting the Restricted Growth Function Genetic Algorithm for Grouping Problems', Vol. 19, No. 4.

22. Jaśkowski, W. and Krawiec, K. (2011) 'Formal Analysis, Hardness, and Algorithms for Extracting Internal Structure of Test-Based Problems', Vol. 19, No. 4.
23. Doerr, B., Happ, E. and Klein, C. (2011) 'Tight Analysis of the (1 + 1)-EA for the Single Source Shortest Path Problem', Vol. 19, No. 4.
24. McClymont, K. and Keedwell, E. (2012) 'Deductive Sort and Climbing Sort: New Methods for Non-Dominated Sorting', Vol. 20, No. 1.
25. Deb, K. and Saha, A. (2012) 'Multimodal Optimization Using a Bi-Objective Evolutionary Algorithm', Vol. 20, No. 1.
26. Burke, E.K., Hyde, M.R., Kendall, G. and Woodward, J. (2012) 'Automating the Packing Heuristic Design Process with Genetic Programming', Vol. 20, No. 1.
27. Mouret, J.-B. and Doncieux, S. (2012) 'Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study', Vol. 20, No. 1.
28. Hauschild, M.W., Pelikan, M., Sastry, K. and Goldberg, D.E. (2012) 'Using Previous Models to Bias Structural Learning in the Hierarchical BOA', Vol. 20, No. 1.
29. Smith, J.E. (2012) 'Estimating Meme Fitness in Adaptive Memetic Algorithms for Combinatorial Problems', Vol. 20, No. 2.
30. Wagner, T. and Wessing, S. (2012) 'On the Effect of Response Transformations in Sequential Parameter Optimization', Vol. 20, No. 2.
31. Shutters, S.T. (2012) 'Punishment Leads to Cooperative Behavior in Structured Societies', Vol. 20, No. 2.
32. Ren, Z., Jiang, H., Xuan, J. and Luo, Z. (2012) 'Hyper-Heuristics with Low Level Parameter Adaptation', Vol. 20, No. 2.
33. Bischl, B., Mersmann, O., Trautmann, H. and Weihs, C. (2012) 'Resampling Methods for Meta-Model Validation with Recommendations for Evolutionary Computation', Vol. 20, No. 2.
34. Morgan, R. and Gallagher, M. (2012) 'Using Landscape Topology to Compare Continuous Metaheuristics: A Framework and Case Study on EDAs and Ridge Structure', Vol. 20, No. 2.
35. Liu, J., Abbass, H.A., Green, D.G. and Zhong, W. (2012) 'Motif Difficulty (MD): A Predictive Measure of Problem Difficulty for Evolutionary Algorithms Using Network Motifs', Vol. 20, No. 3.
36. Sun, J., Fnag, W., Wu, X., Palade, V. and Xu, W. (2012) 'Quantum-Behaved Particle Swarm Optimization: Analysis of Individual Particle Behavior and Parameter Selection', Vol. 20, No. 3.
37. Gutjahr, W.J. (2012) 'Runtime Analysis of an Evolutionary Algorithm for Stochastic Multi-Objective Combinatorial Optimization', Vol. 20, No. 3.
38. Hadka, D. and Reed, P. (2012) 'Diagnostic Assessment of Search Controls and Failure Modes in Many-Objective Evolutionary Optimization', Vol. 20, No. 3.
39. Devert, A., Weise, T. and Tang, K. (2012) 'A Study on Scalable Representations for Evolutionary Optimization of Ground Structures', Vol. 20, No. 3.
40. Bouhmala, N. (2012) 'A Multilevel Memetic Algorithm for Large SAT-Encoded Problems', Vol. 20, No. 4.
41. Pošík, P. and Kubalík, J.(2012) 'Experimental Comparison of Six Population-Based Algorithms for Continuous Black Box Optimization', Vol. 20, No. 4.
42. Pošík, P., Huyer, W. and Pál, L. (2012) 'A Comparison of Global Search Algorithms for Continuous Black Box Optimization', Vol. 20, No. 4.
43. Müller, C.L. and Sbalzarini, I.F. (2012) 'Energy Landscapes of Atomic Clusters as Black Box Optimization Benchmarks', Vol. 20, No. 4.
44. Pošík, P. and Huyer, W. (2012) 'Restarted Local Search Algorithms for Continuous Black Box Optimization', Vol. 20, No. 4.
45. Pál, L., Csendes, T., Markót, M.C. and Neumaier, A. (2012) 'Black Box Optimization Benchmarking of the GLOBAL Method', Vol. 20, No. 4.